Deep hedging

H. Buehler, L. Gonon, J. Teichmann & B. Wood

Quantitative Finance, 2019
https://doi.org/10.1080/14697688.2019.1571683

Deep hedging

H. BUEHLER†[§], L. GONON‡*, J. TEICHMANN‡ and B. WOOD†[§]

†J.P. Morgan, London, UK
‡Eidgenössische Technische Hochschule Zürich, Zürich, Switzerland

(*Received 15 February 2018; accepted 9 January 2019; published online 21 February 2019*)

We present a framework for hedging a portfolio of derivatives in the presence of market frictions such as transaction costs, liquidity constraints or risk limits using modern deep reinforcement machine learning methods. We discuss how standard reinforcement learning methods can be applied to non-linear reward structures, i.e. in our case convex risk measures. As a general contribution to the use of deep learning for stochastic processes, we also show in Section 4 that the set of constrained trading strategies used by our algorithm is large enough to ϵ-approximate any optimal solution. Our algorithm can be implemented efficiently even in high-dimensional situations using modern machine learning tools. Its structure does not depend on specific market dynamics, and generalizes across hedging instruments including the use of liquid derivatives. Its computational performance is largely invariant in the size of the portfolio as it depends mainly on the number of hedging instruments available. We illustrate our approach by an experiment on the S&P500 index and by showing the effect on hedging under transaction costs in a synthetic market driven by the Heston model, where we outperform the standard 'complete-market' solution.

Keywords: Reinforcement learning; Machine learning; Market frictions; Transaction costs; Hedging; Risk management; Portfolio optimization

JEL Classification: C45

1. Introduction

The problem of pricing and hedging portfolios of derivatives is crucial for pricing risk-management in the financial securities industry. In idealized frictionless and 'complete-market' models, mathematical finance provides, with risk-neutral pricing and hedging, a tractable solution to this problem. Most commonly, in such models only the primary asset such as the equity and a few additional factors are modeled. Arguably, the most successful such model for equity models is Dupire's Local Volatility (Dupire 1994), see e.g. Crépey (2004). For risk management, we will then compute 'greeks' with respect not only to spot but also to calibration input parameters such as forward rates and implied volatilities—even if such quantities are not actually state variables in the underlying model. Essentially models are used as a low dimensional interpolation technique of the hedging instruments and the market information from which hedging decisions are derived. Under complete-market assumptions, pricing and risk of a portfolio of derivatives is additionally linear. Of course such approaches are limited and it is in any model-driven decision making process *easy* to imagine market environments where the model-driven decisions actually fail to hedge properly.

In real markets, though, trading in any instrument is subject to transaction costs, permanent market impact and liquidity constraints. Furthermore, any trading desk is typically also limited by its capacity for risk and stress, or more generally capital. This requires traders to overlay the trading strategy implied by the greeks computed from the complete-market model with their own adjustments. It also means that pricing and risk are not linear but dependent on the overall book: a new trade which reduces the risk in a particular direction can be priced more favorably. This phenomenon is called having an 'axe'.

The prevalent use of the complete-market models is due to a lack of efficient alternatives; even with the impressive progress made in the last years for example with respect to robust hedging or super-hedging, there are still few solutions which will scale well over a large portfolio of instruments, and which do not depend on the underlying market dynamics.

Our *deep hedging* approach addresses this deficiency. Essentially, we model the trading decisions in our hedging

*Corresponding author. Email: lukas.gonon@math.ethz.ch
[§]Opinions expressed in this paper are those of the authors, and do not necessarily reflect the view of J.P. Morgan.

strategies as neural networks; their *feature sets* consist not only of prices of our hedging instruments but may also contain additional information such as trading signals, news analytics or past hedging decisions—quantitative information a human trader might use, in a true machine learning fashion.

Such deep hedging strategies can be described and trained (optimized in classical language) in a very efficient way, while the respective algorithms are entirely model-free and do not depend on the chosen market dynamics. That means we can include market frictions such as transaction costs, liquidity constraints, bid/ask spreads, etc., all potentially dependent on the features of the scenario.

The modeling task now amounts to specifying a market scenario generator, a loss function, market frictions and trading instruments. This approach lends itself well to statistically driven market dynamics. That also means that we do not need to compute greeks of individual derivatives with a classic derivative pricing model. In fact, we will need no such 'equivalent martingale measure model': *our approach is greek-free*. Instead, we can focus our modeling effort on realistic market dynamics and the actual out-of-sample performance of our hedging signal.

High level optimizers then find reasonably good strategies to achieve good out-of-sample hedging performance under the stated objective. In our examples, we are using gradient descent 'Adam' (Kingma and Ba 2015) minibatch training for a semi-recurrent reinforcement learning problem.

To illustrate our approach, we will build on ideas from Föllmer and Leukert (2000) and Ilhan *et al.* (2009) and optimize hedging of a portfolio of derivatives under *convex risk measures*. To be able to compare our results with classic complete-market results, we chose in this article to drive the market with a Heston model. To show that the approach is also feasible in practice, these results are complemented by an experiment on the S&P500 index. We re-iterate that our algorithm is not dependent on the choice of the model.

To illustrate our algorithm, we investigate the following questions:

- Section 5.2: How does neural network hedging (for different risk-preferences) compare to the benchmark in a Heston model without transaction costs?
- Section 5.3: What is the effect of proportional transaction costs on the exponential utility indifference price?
- Section 5.4: Is the numerical method scalable to higher dimensions?
- Section 5.5: How does it work in practice, e.g. when compared to a daily-recalibrated Black–Scholes model on the S&P500 index?

Our analysis is based on out-of-sample performance.

To calculate our hedging strategies numerically, we approximate them by deep neural networks. State-of-the-art machine learning optimization techniques (see Goodfellow *et al.* 2016) are then used to train these networks, yielding a close-to-optimal *deep hedge*. This is implemented in Python using TensorFlow. Under our Heston model, trading is allowed in both stock and a variance swap. Even experiments with proportional transaction costs show promising results and the approach is also feasible in a high-dimensional setting.

1.1. Comparison to standard hedging techniques

We want to emphasize that our deep hedging methodology is generic, i.e. for *any* sufficiently rich set of market scenarios we can calculate hedging strategies for *any* pre-specified set of hedging instruments in the presence of *any* sort of market frictions and *any* risk quantification methodology in an efficient way. In contrast to most of the industry applied methods the hedging strategies are *not* calculated within a model framework.

We have compared our methodology to the following standard methodology: 'take a classical stochastic price model, calibrate it to market data and calculate the model hedge for a given payoff and risk quantification methodology'. As a first example we have taken a Heston model and used equity and a variance swap as hedging instruments. We could have chosen here any other setup with known hedging strategies. In other words: the complexity of our methodology is not influenced by the nature of the law on path space. In addition, as a second example we have fitted a standard econometric model to the S&P500 index and evaluated the hedging performance on real data.

Of course, in specific situations, e.g. when hedging vanilla options on a single underlying for which a lot of option price data is available, there are many well-established methods, see e.g. Bates (2005), Alexander and Nogueira (2007), Sepp (2012) and Hull and White (2017). The key contribution of the present work is that the proposed methodology is not limited to such situations. We also want to point out that most of the standard methods do neither take into account market frictions nor trading constraints.

We plan in a future work to quantify first time model risk, i.e. the likely mis-specification of the scenario generation, since we can optimize over different scenario generators due to the generic character of our approach.

1.2. Related literature

There is a vast literature on hedging in market models with frictions. We only highlight a few to demonstrate the complex character of the problem. Hedging and utility indifference pricing under transaction costs has been studied starting with Hodges and Neuberger (1989) and Davis *et al.* (1993). As apparent from these papers, calculating the exponential utility indifference price for a call option even in a Black–Scholes model requires solving a multidimensional non-linear free boundary problem. This has motivated studies on asymptotic behavior of prices and strategies as e.g. in Whalley and Wilmott (1997), Barles and Soner (1998), Kallsen and Muhle-Karbe (2015). For extensive lists of references, which also cover alternative approaches and portfolio choice problems under transaction costs, we refer to Carmona (2009), Kabanov and Safarian (2009), recent contributions as e.g. (Bouchard *et al.* 2016) and the survey (Muhle-Karbe *et al.* 2017).

A closely related setting, which is also covered by our framework, is considered in Rogers and Singh (2010). The authors study a market with quadratic transaction costs, which is interpreted as a temporary price impact. The price process is modeled by a one-dimensional Black–Scholes model. The optimal trading strategy can be obtained by solving a system of three coupled (non-linear) PDEs. In Bank *et al.* (2017) a more general tracking problem is carried out for a Bachelier model and a closed form solution (involving conditional expectations of a time integral over the optimal frictionless hedging strategy) is obtained for the strategy. Soner *et al.* (1995) prove that in a Black–Scholes market with proportional transaction costs, the cheapest superhedging price for a European call option is the spot price of the underlying. Thus, the concept of super-replication is of little interest to practitioners in the one-dimensional case. In higher dimensional cases it suffers from numerical intractability.

It is well known that deep feed forward networks satisfy universal approximation properties, see e.g. Hornik (1991). To understand better why they are so efficient at approximating hedging strategies, we rely on the very recent and fascinating results of Bölcskei *et al.* (2017), which can be stated as follows: they quantify the minimum network connectivity needed to allow approximation of *all* elements in pre-specified classes of functions to within a prescribed error, which establishes a universal link between the connectivity of the approximating network and the complexity of the function class that is approximated. An abstract framework for transferring optimal M-term approximation results with respect to a *representation system* to optimal M-edge approximation results for neural networks is established. These transfer results hold for dictionaries that are *representable by neural networks* and it is also shown in Bölcskei *et al.* (2017) that a wide class of representation systems, coined *affine systems*, and including as special cases wavelets, ridgelets, curvelets, shearlets, α-shearlets, and more generally, α-molecules, as well as tensor-products thereof, are re-presentable by neural networks. These results suggest an explanation for the 'unreasonable effectiveness' of neural networks: they effectively combine the optimal approximation properties of all affine systems taken together. In our application of deep hedging strategies this means: understanding the relevant input factors for which the optimal hedging strategy can be written efficiently.

There are several related applications of reinforcement learning in finance which have similar challenges, of which we want to highlight two related streams: the first is the application to classic portfolio optimization, i.e. without options and under the assumption that market prices are available for all hedging instruments. As in our setup, this problem requires the use of non-linear objective functions, c.f. for example Moody and Wu (1997) or Jiang *et al.* (2017). The second promising application of reinforcement learning is in algorithmic trading, where several authors have shown promising results, e.g. Du *et al.* (2009) and Lu (2017) to give but two examples.

The novelty in this article is that we cover derivatives in the first place, and in particular over-the-counter derivatives which do not have an observable market price. For example, Halperin (2017) covers hedging using Q-learning with only the stock price under Black–Scholes assumptions and without transaction cost. The article proposes a discrete-time hedging methodology with a mean-variance type optimality criterion and aims at learning the action-value function, which is parametrized by a set of basis functions chosen by the user. A key difference to our work is that the approach does not address the question of how to incorporate multiple hedging instruments and market frictions such as e.g. transaction costs.

Finally, let us point out that if a derivative is very liquidly traded and a large quantity of historical price data is available, then an alternative approach first proposed in Hutchinson *et al.* (1994) is to approximate the pricing function of the derivative by a neural network and use a hedging strategy based on the greeks of the optimized neural network pricing function. The deep hedging approach proposed here directly parametrizes the hedging strategy and can thus be applied also in situations in which none or only very little price data of the derivative to be hedged is available. In addition, the deep hedging methodology allows to incorporate multiple hedging instruments and market frictions such as e.g. transaction costs.

This puts our article firmly in the realm of pricing and risk managing a contingent claims in incomplete markets with friction cost. A general introduction into quantitative finance with a focus on such markets is Föllmer and Schied (2016).

1.3. Outline

The rest of the article is structured as follows. In Sections 2 and 3 we provide the theoretical framework for pricing and hedging using convex risk measures in discrete-time markets with frictions. Section 4 outlines the parametrization of appropriate hedging strategies by neural nets and provides theoretical arguments why it works. In Section 5 several numerical experiments are performed demonstrating the surprising feasibility and accuracy of the method.

2. Setting: discrete-time market with frictions

Consider a discrete-time financial market with finite time horizon T and trading dates $0 = t_0 < t_1 < \ldots < t_n = T$. Fix a finite† probability space $\Omega = \{\omega_1, \ldots, \omega_N\}$ and a probability measure $\mathbb{P}$ such that $\mathbb{P}[\{\omega_i\}] > 0$ for all i. We define the set of all real-valued random variables over Ω as $\mathcal{X} := \{X : \Omega \to \mathbb{R}\}$.

We denote by I_k with values in $\mathbb{R}^r$ any new market information available at time t_k, including market costs and mid-prices of liquid instruments—typically quoted in auxiliary terms such as implied volatilities—news, balance sheet information, any trading signals, risk limits etc. The process $I = (I_k)_{k=0,\ldots,n}$ generates the filtration $\mathbb{F} = (\mathcal{F}_k)_{k=0,\ldots,n}$, i.e. $\mathcal{F}_k$ represents all information available up to t_k. Note that each $\mathcal{F}_k$-measurable random variable can be written as a function of $I_0, \ldots, I_k$; this is therefore the richest available feature set for any decision taken at t_k.

The market contains d hedging instruments with mid-prices given by an $\mathbb{R}^d$-valued $\mathbb{F}$-adapted stochastic process $S = (S_k)_{k=0,\ldots,n}$. We do *not* require that there is an equivalent martingale measure under which S is a martingale. We stress

† The assumption that Ω is finite is only essential for the numerical solution of the optimal hedging problem (from Section 4.3 onwards). Alternatively, we could start with arbitrary Ω and discretize it for the numerical solution. If we imposed appropriate integrability conditions on all assets and contingent claims, then the results prior to Section 4.3 would remain valid for general Ω.

that our hedging instruments are not simply primary assets such as equities but also secondary assets such as liquid options on the former. Some of those hedging instruments are therefore not tradable before a future point in time (e.g. an option only listed in 3M with then time-to-maturity of 6M). Such liquidity restrictions are modeled alongside trading cost below.

Our portfolio of derivatives which represents our liabilities is an $\mathcal{F}_T$ measurable random variable Z. In keeping with the classic literature we may refer to this as the *contingent claim* but we stress that it is meant to represent a portfolio which is a mix of liquid and OTC derivatives. The maturity T is the maximum maturity of all instruments, at which point all payments are known.

No classic derivative pricing model will be needed to valuate Z or compute Greeks at any point.

2.1. *Simplifications*

For notational simplicity, we assume that all intermediate payments are accrued using a (locally) risk-free overnight rate. This essentially means we may assume that rates are zero and that all payments occur at T. We also exclude for the purpose of this article instruments with true optionality such as American options. Finally, we also assume that all currency spot exchange happens at zero cost, and that we therefore may assume that all instruments settle in our reference currency.†

2.2. *Trading strategies*

In order to hedge a liability Z at T, we may trade in S using an $\mathbb{R}^d$-valued $\mathbb{F}$-adapted stochastic process $\delta = (\delta_k)_{k=0,\ldots,n-1}$ with $\delta_k = (\delta_k^1, \ldots, \delta_k^d)$. Here, δ_k^i denotes the agent's holdings of the ith asset at time t_k. We may also define $\delta_{-1} = \delta_n := 0$ for notational convenience.

We denote by $\mathcal{H}^u$ the unconstrained set of such trading strategies. However, each δ_k is subject to additional trading constraints. Such restrictions arise due to liquidity, asset availability or trading restrictions. They are also used to restrict trading in a particular option prior to its availability. In the example above of an option which is listed in 3M, the respective trading constraints would be $\{0\}$ until the 3M point. To incorporate these effects, we assume that δ_k is restricted to a set $\mathcal{H}_k$ which is given as the image of a continuous, $\mathcal{F}_k$-measurable map $H_k : \mathbb{R}^{d(k+1)} \to \mathbb{R}^d$, i.e. $\mathcal{H}_k := H_k(\mathbb{R}^{d(k+1)})$. We stipulate that $H_k(0) = 0$.

Moreover, for an unconstrained strategy $\delta^u \in \mathcal{H}^u$, we (successively) define with $(H \circ \delta^u)_k := H_k((H \circ \delta^u)_0, \ldots, (H \circ \delta^u)_{k-1}, \delta_k^u)$ its constrained 'projection' into $\mathcal{H}_k$. We denote by $\mathcal{H} := (H \circ \mathcal{H}^u) \subset \mathcal{H}^u$ the corresponding non-empty set of restricted trading strategies.

Example 1 Assume that S are a range of options and that $\mathcal{V}_k^i(S_k^i)$ computes the Black–Scholes Vega of each option using the various market parameters available at time t_k. The overall Vega traded with δ_k is then $\mathcal{V}_k(\delta_k - \delta_{k-1}) := |\sum_{i=1}^d \mathcal{V}_k^i(S_k^i)(\delta_k^i - \delta_{k-1}^i)|$. A liquidity limit of a maximum tradable Vega of $\mathcal{V}_{\max}$ could then be implemented by the map:

$$H_k(\delta_0, \ldots, \delta_k) := \delta_{k-1} + (\delta_k - \delta_{k-1}) \times \frac{\mathcal{V}_{\max}}{\max\{\mathcal{V}_k(\delta_k - \delta_{k-1}), \mathcal{V}_{\max}\}}.$$

2.3. *Hedging*

All trading is self-financed, so we may also need to inject additional cash p_0 into our portfolio. A negative cash injection implies we may extract cash. In a market without transaction costs the agent's wealth at time T is thus given by $-Z + p_0 + (\delta \cdot S)_T$, where

$$(\delta \cdot S)_T := \sum_{k=0}^{n-1} \delta_k \cdot (S_{k+1} - S_k).$$

However, we are interested in situations where trading cost cannot be neglected. We assume that any trading activity causes costs as follows: if the agent decides to buy a position $n \in \mathbb{R}^d$ in S at time t_k, then this will incur cost $c_k(n)$. The total cost of trading a strategy δ up to maturity is therefore

$$C_T(\delta) := \sum_{k=0}^{n} c_k(\delta_k - \delta_{k-1})$$

(recall $\delta_{-1} = \delta_n := 0$, the latter of which implies full liquidation in T). The agent's terminal portfolio value at T is therefore

$$\mathrm{PL}_T(Z, p_0, \delta) := -Z + p_0 + (\delta \cdot S)_T - C_T(\delta). \tag{1}$$

Throughout, we assume that the non-negative adapted cost functions are normalized to $c_k(0) = 0$ and that they are upper semi-continuous.‡ In our numerical examples we have assumed zero transaction costs at maturity.

Our setup includes the following effects:

- Proportional transaction cost: for for $c_k^i > 0$ define $c_k(n) := \sum_{i=1}^d c_k^i S_k^i |n^i|$.
- Fixed transaction costs: for $c_k^i > 0$ and $\varepsilon > 0$ set $c_k(n) := \sum_{i=1}^d c_k^i 1_{|n^i| \geq \varepsilon}$.
- Complex cross-asset cost, such as cost of volatility when trading options across the surface: assume S^1 is spot and that the rest of the hedging instruments are options on the same asset. Denote by Δ_k^i Delta and by $\mathcal{V}_k^i$ Vega of each instrument, for example under a simple Black–Scholes model.We may then define a simple cross-surface proportional cost model in Delta and Vega for $c_k > 0$ and $v_k > 0$ as

$$c_k(n) := c_k^i S_k^1 \left| 1 + \sum_{i=2}^d \Delta_k^i n^i \right| + v_k^i \left| \sum_{i=2}^d \mathcal{V}_k^i n^i \right|.$$

Remark 1 We believe that our general setup can also be extended to include true market impact: in this case, the asset distribution is affected by our trading decisions.

† See Burgert and Rüschendorf (2006) for some background on multi-currency risk measures.

‡ This property is needed in the proof of proposition 4.3.

As an example for permanent market impact, assume for simplicity that $I = S$ and that we have a statistical model of our market in the form of a conditional distribution $P(S_{k+1}|S_k)$. For a proportional impact parameter $\iota > 0$ we may now define the dynamics of S under exponentially decaying, proportional market impact as $P(\ S_{k+1} \mid S_k(1 + \iota(\delta_k - \delta_{k-1}))\ )$. The cost function is accordingly $c_k(n) := S_k \iota |n|$.

In a similar vein, dynamic market impact with decay such as described in Gatheral and Schied (2013) can be implemented.

The real challenge with modeling impact is the effect of trading in one hedging instrument on other hedging instruments, for example when trading options.

Remark 2 We have chosen here to formulate the problem and our analysis in the language of mathematical finance rather than reinforcement learning. This could be translated by interpreting market information as states, trading strategies as actions and the portfolio value at each time point as a reward and by employing a convex risk measure (as introduced in the next section) as a risk-adjusted return measure.

3. Pricing and hedging using convex risk measures

In an idealized complete market with continuous-time trading, no transaction costs and unconstrained hedging, for any liabilities Z there exists a unique replication strategy δ and a fair price $p_0 \in \mathbb{R}$ such that $-Z + p_0 + (\delta \cdot S)_T - C_T(\delta) = 0$ holds $\mathbb{P}$-a.s. This is not true in our current setting.

In an incomplete market with frictions, an agent has to specify an optimality criterion which defines an acceptable ‘minimal price’ for any position. Such a minimal price is the going to be the minimal amount of cash we need to add to our position in order to implement the optimal hedge and such that the overall position becomes acceptable in light of the various costs and constraints.

We focus here on optimality under *convex risk measures* as studied e.g. in Xu (2006) and Ilhan *et al.* (2009). See also Klöppel and Schweizer (2007) and further references therein for a dynamic setting. Convex risk measures are discussed in great detail in Föllmer and Schied (2016).

Definition 1 Assume that $X, X_1, X_2 \in \mathcal{X}$ represent asset positions (i.e. $-X$ is a liability).

We call $\rho : \mathcal{X} \to \mathbb{R}$ a *convex risk measure* if it is:

(i) Monotone decreasing: if $X_1 \geq X_2$ then $\rho(X_1) \leq \rho(X_2)$. *A more favorable position requires less cash injection.*

(ii) Convex: $\rho(\alpha X_1 + (1-\alpha)X_2) \leq \alpha\rho(X_1) + (1-\alpha)\rho(X_2)$ for $\alpha \in [0,1]$. *Diversification works.*

(iii) Cash-Invariant: $\rho(X + c) = \rho(X) - c$ for $c \in \mathbb{R}$. *Adding cash to a position reduces the need for more by as much. In particular, this means that $\rho(X + \rho(X)) = 0$, i.e. $\rho(X)$ is the least amount c that needs to be added to the position X in order to make it acceptable in the sense that $\rho(X + c) \leq 0$.*

We call ρ *normalized* if $\rho(0) = 0$.

Let $\rho \colon \mathcal{X} \to \mathbb{R}$ be such a convex risk measure and for $X \in \mathcal{X}$ consider the optimization problem

$$\pi(X) := \inf_{\delta \in \mathcal{H}} \rho\ (X + (\delta \cdot S)_T - C_T(\delta))\,. \tag{2}$$

Proposition 3.1 *π is monotone decreasing and cash-invariant.*

If moreover $C_T(\cdot)$ and $\mathcal{H}$ are convex, then the functional π is a convex risk measure.

Proof For convexity, let $\alpha \in [0,1]$, set $\alpha' := 1 - \alpha$ and assume $X_1, X_2 \in \mathcal{X}$. Then using the definition of π in the first step, convexity of $\mathcal{H}$ in the second step, convexity of $C_T(\cdot)$ combined with monotonicity of ρ in the third step and convexity of ρ in the fourth step, we obtain

$$\begin{aligned}
&\pi(\alpha X_1 + \alpha' X_2)\\
&\quad = \inf_{\delta \in \mathcal{H}} \rho\left(\alpha X_1 + \alpha' X_2 + (\delta \cdot S)_T - C_T(\delta)\right)\\
&\quad = \inf_{\delta_1,\delta_2 \in \mathcal{H}} \rho\big(\alpha\left\{X_1 + (\delta_1 \cdot S)_T\right\} + \alpha'\left\{X_2 + (\delta_2 \cdot S)_T\right\}\\
&\quad -C_T(\alpha\delta_1 + \alpha'\delta_2)\big)\\
&\quad \leq \inf_{\delta_1,\delta_2 \in \mathcal{H}} \rho\left(\alpha\left\{X_1 + (\delta_1 \cdot S)_T - C_T(\delta_1)\right\}\right.\\
&\quad \left.+\alpha'\left\{X_2 + (\delta_2 \cdot S)_T - C_T(\delta_2)\right\}\right)\\
&\quad \leq \inf_{\delta_1,\delta_2 \in \mathcal{H}} \left\{\alpha\rho\left(X_1 + (\delta_1 \cdot S)_T - C_T(\delta_1)\right)\right.\\
&\quad \left.+\alpha'\rho\left(X_2 + (\delta_2 \cdot S)_T - C_T(\delta_2)\right)\right\}\\
&\quad = \alpha\pi(X_1) + \alpha'\pi(X_2).
\end{aligned}$$

Cash invariance and monotonicity follow directly from the respective properties of ρ. ■

We define an optimal hedging strategy as a minimizer $\delta \in \mathcal{H}$ of (2). Recalling the interpretation of $\rho(-Z)$ as the minimal amount of capital that has to be added to the risky position $-Z$ to make it acceptable for the risk measure ρ, this means that $\pi(-Z)$ is simply the minimal amount that the agent needs to charge in order to make her terminal position acceptable, if she hedges optimally.

If we defined this as the minimal price, then we would exclude the possibility that having no liabilities may actually have positive value. This might be the case in the presence of statistically positive expectation of returns under $\mathbb{P}$ for some of our hedging instruments. As mentioned before, our framework lends itself to the integration of signals and other trading information. We therefore define the *indifference price* $p(Z)$ as the amount of cash that she needs to charge in order to be indifferent between the position $-Z$ and not doing so, i.e. as the solution p_0 to $\pi(-Z + p_0) = \pi(0)$. By cash invariance this is equivalent to taking $p_0 := p(Z)$, where

$$p(Z) := \pi(-Z) - \pi(0). \tag{3}$$

It is easily seen that without trading restrictions and transaction costs, this price coincides with the price of a replicating portfolio (if it exists):

Lemma 3.2 *Suppose $C_T \equiv 0$ and $\mathcal{H} = \mathcal{H}^u$. If Z is attainable, i.e. there exists $\delta^* \in \mathcal{H}$ and $p_0 \in \mathbb{R}$ such that $Z = p_0 + (\delta^* \cdot S)_T$, then $p(Z) = p_0$.*

Proof For any $\delta \in \mathcal{H}$, the assumptions and cash invariance of ρ imply

$$\rho\left(-Z + (\delta \cdot S)_T - C_T(\delta)\right) = p_0 + \rho(([\delta - \delta^*] \cdot S)_T).$$

Taking the infimum over $\delta \in \mathcal{H}$ on both sides and using $\mathcal{H} - \delta^* = \mathcal{H}$ one obtains

$$\pi(-Z) = p_0 + \inf_{\delta \in \mathcal{H}} \rho(([\delta - \delta^*] \cdot S)_T) = p_0 + \pi(0). \qquad ■$$

Remark 3 The methodology developed in this article can also be applied to approximate optimal hedging strategies in a setting where the price p_0 is given exogenously: fix a loss function $\ell\colon \mathbb{R} \to [0, \infty)$. Suppose $p_0 > 0$ is given, for example being the result of trading derivatives in the market at competitive prices, without taking into account risk-management. The agent then wishes to minimize her loss at maturity, i.e. she defines an optimal hedging strategy as a minimizer to

$$\inf_{\delta \in \mathcal{H}} \mathbb{E}\left[\, \ell(-Z + p_0 + (\delta \cdot S)_T - C_T(\delta))\,\right]. \tag{4}$$

This problem, i.e. optimal hedging under a capital constraint, is closely related to taking for ρ a shortfall risk measure, see e.g. Föllmer and Leukert (2000).

3.1. Arbitrage

We mentioned in the introduction that we do not require per se that the market is free of arbitrage. To recap, we call $\delta^{[X]} \in \mathcal{H}$ an arbitrage opportunity given X is an opportunity to make money without risk of a loss, i.e. $0 \le X + (\delta^{[X]} S)_T - C_T(\delta^{[X]}) =: (*)$ while $\mathbb{P}[(*) > 0] > 0$.

In case such an opportunity exists, we obviously have $\rho(X) < 0$. Depending on the cost function and our constraints $\mathcal{H}$, we may be able to invest an unlimited amount into this strategy. In this case, we get $\pi(X) = -\infty$. If this applies to $X = 0$, we call such a market *irrelevant*. This is justified by the following observation:

Corollary 3.3 *Assume that $\pi(0) > -\infty$. Then $\pi(X) > -\infty$ for all X.*

Proof Since Ω is finite we have $\sup X < \infty$ and therefore, using monotonicity, $\pi(X) \ge \pi(\sup X) \ge \pi(0) - \sup X > -\infty$. ■

We note, however, that irrelevance is not necessarily a consequence of outright arbitrage; such *statistical arbitrage* may also occur in markets without arbitrage. Consider to this end the convex risk measure $\rho(X) := -\mathbb{E}[X]$, and assume that the market without interest rates is driven by a standard Black–Scholes model with positive drift μ between two time points t_0 and t_1, i.e.

$$S_0 := 1 \quad \text{and} \quad S_1 := \exp\left\{\mu t_1 + \sigma Z \sqrt{t_1}\right\}$$

for Z normal and a volatility $\sigma > 0$. Assume the proportional cost of trading S in t_0 is $0.5\mathrm{e}^{\mu t_1}$. In this case $\rho(\delta_0 S_1 - C_0(\delta)) = -0.5\delta_0 \mathrm{e}^{\mu t_1}$ for any $\delta_0 \in \mathbb{R}$ which implies $\pi(0) = -\infty$. Hence, the market is irrelevant, too, even if it does not exhibit classic arbitrage. We also note that this is expected in practise: as an example, consider a strategy which writes options on an underlying. In most market scenarios such a strategy will on average make money, even if it is subject to potentially drastic short-term losses.

In closing we note that even if the market dynamics exhibit classic arbitrage, and even in the absence of cost or liquidity constraints, we may not be able to exploit it. Let us assume that for every arbitrage opportunity $\delta^{[0]}$ there is a non-zero probability of not making money, i.e. $\mathbb{P}[(\delta^{[0]} S)_T + C_T(\delta^{[0]}) = 0] > 0$. Under the extreme risk measure $\rho(X) := -\inf X$ this market remains relevant with $\pi(0) = 0$.

3.2. Exponential utility indifference pricing

The following lemma shows that the present framework includes exponential utility indifference pricing as studied for example in Hodges and Neuberger (1989), Davis *et al.* (1993), Whalley and Wilmott (1997) and Kallsen and Muhle-Karbe (2015). Recall that for the exponential utility function $U(x) := -\exp(-\lambda x), x \in \mathbb{R}$ with risk-aversion parameter $\lambda > 0$ the indifference price $q(Z) \in \mathbb{R}$ of Z is defined by

$$\begin{aligned}&\sup_{\delta \in \mathcal{H}} \mathbb{E}\left[U(q(Z) - Z + (\delta \cdot S)_T + C_T(\delta))\right] \\ &\quad = \sup_{\delta \in \mathcal{H}} \mathbb{E}\left[U((\delta \cdot S)_T + C_T(\delta))\right].\end{aligned}$$

In other words, if the seller charges a cash amount of $q(Z)$, sells Z and trades in the market, she obtains the same expected utility as by not not selling Z at all.

Lemma 3.4 *Define $q(Z)$ as above. Choose ρ as the entropic risk measure*

$$\rho(X) = \frac{1}{\lambda} \log \mathbb{E}[\exp(-\lambda X)], \tag{5}$$

and define $p(Z)$ by (3). Then $q(Z) = p(Z)$.

Proof Using the special form of U, one may write the indifference price as

$$q(Z) = \frac{1}{\lambda} \log\left(\frac{\sup_{\delta \in \mathcal{H}} \mathbb{E}\left[U(-Z + (\delta \cdot S)_T + C_T(\delta))\right]}{\sup_{\delta \in \mathcal{H}} \mathbb{E}\left[U((\delta \cdot S)_T + C_T(\delta))\right]}\right)$$

and so the claim follows from (3) and (5). ■

3.3. Optimized certainty equivalents

Assume that $\ell\colon \mathbb{R} \to \mathbb{R}$ is a *loss function*, i.e. continuous, non-decreasing and convex. We may define a convex risk measure ρ by setting

$$\rho(X) := \inf_{w \in \mathbb{R}} \left\{w + \mathbb{E}[\ell(-X - w)]\right\}, \quad X \in \mathcal{X}. \tag{6}$$

Lemma 3.5 *(6) defines a convex risk measure.*

Proof Let $X, Y \in \mathcal{X}$ be assets.

(i) Monotonicity: suppose $X \le Y$. Since ℓ is non-decreasing, for any $w \in \mathbb{R}$ one has $\mathbb{E}[\ell(-X-w)] \ge \mathbb{E}[\ell(-Y-w)]$ and thus $\rho(X) \ge \rho(Y)$.

(ii) Cash invariance: for any $m \in \mathbb{R}$, (6) gives

$$\rho(X+m) = \inf_{w\in\mathbb{R}} \{(w+m) - m + \mathbb{E}$$
$$[\ell(-X-(w+m))]\} = -m + \rho(X).$$

(iii) Convexity: let $\lambda \in [0,1]$. Then convexity of ℓ implies

$$\begin{aligned}
&\rho(\lambda X + (1-\lambda)Y)\\
&\quad = \inf_{w\in\mathbb{R}} \{w + \mathbb{E}[\ell(-\lambda X - (1-\lambda)Y - w)]\}\\
&\quad = \inf_{w_1,w_2\in\mathbb{R}} \{\lambda w_1 + (1-\lambda)w_2\\
&+\mathbb{E}[\ell(\lambda(-X-w_1) + (1-\lambda)(-Y-w_2))]\}\\
&\quad \le \inf_{w_1\in\mathbb{R}} \inf_{w_2\in\mathbb{R}} \{\lambda(w_1 + \mathbb{E}[\ell(-X-w_1)])\\
&+(1-\lambda)(w_2 + \mathbb{E}[\ell(-Y-w_2)])\}\\
&\quad = \lambda\rho(X) + (1-\lambda)\rho(Y).
\end{aligned}$$

■

Taking $\ell(x) := -u(-x)$ $(x \in \mathbb{R})$ for a utility function $u\colon \mathbb{R} \to \mathbb{R}$, (6) coincides with the optimized certainty equivalent as defined (and studied in a lot more detail than here) in Ben-Tal and Teboulle (2007).

Example 2 Fix $\lambda > 0$ and set $\ell(x) := \exp(\lambda x) - ((1 + \log(\lambda))/\lambda)$, $x \in \mathbb{R}$. Then the optimization problem in (6) can be solved explicitly and the minimizer w^* satisfies $e^{\lambda w^*} = \lambda\mathbb{E}[\exp(-\lambda X)]$. Inserting this into (6), one obtains the *entropic risk measure* defined in (5) above.

Example 3 Let $\alpha \in (0,1)$ and set $\ell(x) := (1/(1-\alpha)) \max(x, 0)$. The associated risk measure (6) is called *average value at risk at level* $1-\alpha$ (see Föllmer and Schied 2016, Definition 4.48, Proposition 4.51 with $\lambda := 1-\alpha$) or also *conditional value at risk* or *expected shortfall*.

Proposition 3.6 *Suppose S is a $\mathbb{P}$-martingale, ρ is defined as in (6) and π, p as in (2), (3). Then*

(i) $\pi(0) = \rho(0)$,
(ii) $p(Z) \ge \mathbb{E}[Z]$ *for any* $Z \in \mathcal{X}$.

Proof Since $0 \in \mathcal{H}$ and $C_T(0) = 0$, one has $\pi(0) \le \rho(0)$ for any choice of risk measure ρ in (2). Under the present assumptions the converse inequality is also true: Since S is a martingale, it holds that

$$\mathbb{E}[(\delta\cdot S)_T] = \sum_{j=0}^{n-1} \mathbb{E}\left[\delta_j \mathbb{E}[S_{j+1} - S_j | \mathcal{F}_j]\right] = 0 \quad \text{for any } \delta \in \mathcal{H}. \tag{7}$$

By first applying Jensen's inequality (recall that ℓ is convex) and then using (7), that $C_T(\delta) \ge 0$ for any $\delta \in \mathcal{H}$ and that ℓ is non-decreasing, one obtains

$$\begin{aligned}
\pi(-Z) &= \inf_{w\in\mathbb{R}} \inf_{\delta\in\mathcal{H}} \{w + \mathbb{E}[\ell(Z - (\delta\cdot S)_T + C_T(\delta) - w)]\}\\
&\ge \inf_{w\in\mathbb{R}} \inf_{\delta\in\mathcal{H}} \{w + \ell(\mathbb{E}[Z - (\delta\cdot S)_T + C_T(\delta) - w])\}\\
&\ge \inf_{w\in\mathbb{R}} \{w + \ell(\mathbb{E}[Z] - w)\} = \rho(-\mathbb{E}[Z])\\
&= \mathbb{E}[Z] + \rho(0).
\end{aligned} \tag{8}$$

Inserting $Z = 0$ yields the converse inequality $\pi(0) \ge \rho(0)$ and thus (i). Combining (i), (3) and (8) then directly gives (ii). ■

4. Approximating hedging strategies by deep neural networks

The key idea that we pursue in this article is to approximate hedging strategies by neural networks. Before describing this approach in more detail we recall the definition and approximation properties of neural networks and prove some basic results on hedging strategies built from them. While these results show that the approach is theoretically well-founded, they are only one reason why we have used neural networks (and not some other parametric family of functions) to approximate hedging strategies. Equally important is that optimal hedging strategies built from neural networks can numerically be calculated very efficiently. This is explained first for the case of OCE-risk measures and for entropic risk. Finally, an extension to general risk measures is presented.

For the reader's convenience we start by providing a brief summary of notation: given a liability $-Z$ the goal is to calculate the indifference price $p(Z)$ defined in (3) and find a strategy $\delta \in \mathcal{H}$ that achieves the infimum in $\pi(-Z)$, where π is defined in (2). In the optimization problem (2) and in what follows ρ is a convex risk measure and $\mathcal{H}$ denotes the set of (possibly restricted) trading strategies. Furthermore, for a strategy $\delta \in \mathcal{H}$ the random variables $(\delta\cdot S)_T$ and $C_T(\delta)$ denote the cumulative hedging gains and transaction costs incurred when trading according to δ from today until maturity T, respectively. We refer to Section 2 for precise assumptions and specifications of these quantities. For examples of risk measures and basic properties of ρ, $p(Z)$ and π we refer to Section 3.

4.1. Universal approximation by neural networks

Let us first recall the definition of a (feed forward) neural network:

Definition 2 Let $L, N_0, N_1, \ldots, N_L \in \mathbb{N}$ with $L \ge 2$, let $\sigma\colon \mathbb{R} \to \mathbb{R}$ and for any $\ell = 1, \ldots, L$, let $W_\ell\colon \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ an affine function. A function $F\colon \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ defined as

$$F(x) = W_L \circ F_{L-1} \circ \cdots \circ F_1 \text{with } F_\ell = \sigma \circ W_\ell$$
$$\text{for } \ell = 1, \ldots, L-1$$

is called a (feed forward) neural network. Here the *activation function* σ is applied componentwise. L denotes the number of layers, $N_1, \ldots, N_{L-1}$ denote the dimensions of the hidden layers and N_0, N_L of the input and output layers, respectively. For any $\ell = 1, \ldots, L$ the affine function W_ℓ is given as $W_\ell(x) = A^\ell x + b^\ell$ for some $A^\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and $b^\ell \in \mathbb{R}^{N_\ell}$.

For any $i = 1, \dots N_\ell, j = 1, \dots, N_{\ell-1}$ the number A^ℓ_{ij} is interpreted as the weight of the edge connecting the node i of layer $\ell - 1$ to node j of layer ℓ. The number of non-zero weights of a network is the sum of the number of non-zero entries of the matrices A^ℓ, $\ell = 1, \dots, L$ and vectors b^ℓ, $\ell = 1, \dots, L$.

Denote by $\mathcal{NN}^\sigma_{\infty,d_0,d_1}$ the set of neural networks mapping from $\mathbb{R}^{d_0} \to \mathbb{R}^{d_1}$ and with activation function σ. The next result (Hornik 1991, Theorems 1 and 2) illustrates that neural networks approximate multivariate functions arbitrarily well.

Theorem 4.1 Universal approximation, Hornik (1991)
Suppose σ is bounded and non-constant. The following statements hold:

- *For any finite measure μ on $(\mathbb{R}^{d_0}, \mathcal{B}(\mathbb{R}^{d_0}))$ and $1 \le p < \infty$, the set $\mathcal{NN}^\sigma_{\infty,d_0,1}$ is dense in $L^p(\mathbb{R}^{d_0}, \mu)$.*
- *If in addition $\sigma \in C(\mathbb{R})$, then $\mathcal{NN}^\sigma_{\infty,d_0,1}$ is dense in $C(\mathbb{R}^{d_0})$ for the topology of uniform convergence on compact sets.*

Since each component of an $\mathbb{R}^{d_1}$-valued neural network is an $\mathbb{R}$-valued neural network, this result easily generalizes to $\mathcal{NN}^\sigma_{\infty,d_0,d_1}$ with $d_1 > 1$, see also the beginning of Section 2 in Hornik (1991). Also note that in Hornik (1991) the results are formulated for $L=2$, that is, only one hidden layer is considered and the dimension N_1 of this layer is allowed to become arbitrarily large. Since $\mathcal{NN}^\sigma_{\infty,d_0,1}$ contains this smaller class of networks, Hornik (1991) implies the result as formulated here. A variety of other results with different assumptions on σ or emphasis on approximation rates are available, see e.g. Bölcskei *et al.* (2017) for further references. We point out that in particular, the result stated here is only qualitative and does not provide information on advantages of deep networks over shallow ones (i.e. why $L > 2$ may be preferable). More light on this is shed e.g. in Shaham *et al.* (2018): there it is shown by means of harmonic analysis that non-linear functions which are given on a lower dimensional training data set can be efficiently approximated by deep neural networks. This is highly relevant in our setting since prices of hedging instruments will show dependencies which in turn means that they concentrate with high probability on a relatively low dimensional set in the state space of prices. It is sufficient to approximate the hedging strategies on this low dimensional subset, which by, e.g. Bölcskei *et al.* (2017) or Shaham *et al.* (2018) is efficiently possible.

In what follows, we fix an activation function σ and omit it in the notation, i.e. we write $\mathcal{NN}_{\infty,d_0,d_1} := \mathcal{NN}^\sigma_{\infty,d_0,d_1}$. Furthermore, we denote by $\{\mathcal{NN}_{M,d_0,d_1}\}_{M\in\mathbb{N}}$ a sequence of subsets of $\mathcal{NN}_{\infty,d_0,d_1}$ with the following properties:

- $\mathcal{NN}_{M,d_0,d_1} \subset \mathcal{NN}_{M+1,d_0,d_1}$ for all $M \in \mathbb{N}$,
- $\bigcup_{M\in\mathbb{N}} \mathcal{NN}_{M,d_0,d_1} = \mathcal{NN}_{\infty,d_0,d_1}$,
- for any $M \in \mathbb{N}$, one has $\mathcal{NN}_{M,d_0,d_1} = \{F^\theta : \theta \in \Theta_{M,d_0,d_1}\}$ with $\Theta_{M,d_0,d_1} \subset \mathbb{R}^q$ for some $q \in \mathbb{N}$ (depending on M).

In addition it will be convenient to assume that for any $M \in \mathbb{N}$, $d, d', d_1 \in \mathbb{N}$ with $d' > d$ the set $\mathcal{NN}_{M,d,d_1}$ can be recovered from $\mathcal{NN}_{M,d',d_1}$ by considering only those networks with zero weight connections at the last $d' - d$ input nodes, i.e. with $A^1_{ij} = 0$ for $j > d$ and all i.

Remark 4 We have two classes of examples in mind: the first one is to take for $\mathcal{NN}_{M,d_0,d_1}$ the set of all neural networks in $\mathcal{NN}_{\infty,d_0,d_1}$ with an arbitrary number of layers and nodes, but at most M non-zero weights. The second one is to take for $\mathcal{NN}_{M,d_0,d_1}$ the set of all neural networks in $\mathcal{NN}_{\infty,d_0,d_1}$ with a *fixed architecture*, i.e. a fixed number of layers $L^{(M)}$ and fixed input and output dimensions for each layer. These are specified by d_0, d_1 and some non-decreasing sequences $\{L^{(M)}\}_{M\in\mathbb{N}}$ and $\{N^{(M)}_1\}_{M\in\mathbb{N}}, \dots, \{N^{(M)}_{L^{(M)}-1}\}_{M\in\mathbb{N}}$. In both cases the set $\mathcal{NN}_{M,d_0,d_1}$ is parametrized by matrices A^ℓ and vectors b^ℓ.

4.2. *Optimal hedging using deep neural networks*

Motivated by the universal approximation results stated above, we now consider neural network hedging strategies. Let our activation function therefore be bounded and non-constant.

In order to apply our Theorem 4.1, we represent the optimization over constrained trading strategies $\delta \in \mathcal{H}$ as an optimization over $\delta \in \mathcal{H}^u$ with a following modified objective.

Lemma 4.2 *We may write the constrained problem (2) as the modified unconstrained problem as*

$$\pi(X) = \inf_{\delta\in\mathcal{H}^u} \rho\left(X + (H\circ\delta\cdot S)_T - C_T(H\circ\delta)\right). \quad (3.1')$$

Proof Note that by definition any $\delta^c \in \mathcal{H}$ is of the form $H \circ \delta = \delta^c$ for some $\delta \in \mathcal{H}^u$ (and vice versa) and so (3.1') directly follows from (2). ■

Note that H is typically not one-to-one (e.g. it could be a projection), so there may not be a unique optimizer for (3.1').

Recall that the information available in our market at t_k is described by the observed maximal feature set $I_0, \dots, I_k$. Our trading strategies should therefore depend on this information and on our previous position in our tradable assets. This gives rise to the following semi-recurrent deep neural network structure for our unconstrained trading strategies:

$$\begin{aligned}\mathcal{H}_M &= \{(\delta_k)_{k=0,\dots,n-1} \in \mathcal{H}^u : \delta_k = F_k(I_0,\dots,I_k,\delta_{k-1}),\\ &\quad F_k \in \mathcal{NN}_{M,r(k+1)+d,d}\}\\ &= \{(\delta^\theta_k)_{k=0,\dots,n-1} \in \mathcal{H}^u : \delta^\theta_k = F^{\theta_k}(I_0,\dots,I_k,\delta_{k-1}),\\ &\quad \theta_k \in \Theta_{M,r(k+1)+d,d}\} \end{aligned} \quad (9)$$

We now replace the set $\mathcal{H}^u$ in (3.1') by $\mathcal{H}_M \subset \mathcal{H}^u$. We aim at calculating

$$\begin{aligned}\pi^M(X) &:= \inf_{\delta\in\mathcal{H}_M} \rho(X + (H\circ\delta\cdot S)_T - C_T(H\circ\delta))\\ &= \inf_{\theta\in\Theta_M} \rho(X + (H\circ\delta^\theta\cdot S)_T - C_T(H\circ\delta^\theta)), \end{aligned} \quad (10)$$

where $\Theta_M = \prod_{k=0}^{n-1} \Theta_{M,r(k+1)+d,d}$. Thus, the infinite-dimensional problem of finding an optimal hedging strategy is reduced to the finite-dimensional constraint problem of finding optimal parameters for our neural network.

Remark 5 Our setup becomes truly 'recurrent' if we enforce $\theta^k = \theta^0$ for all k and add 'k' as a parameter into the network. Below proof applies with few modifications.

Remark 6 If S is an $(\mathbb{F}, \mathbb{P})$-Markov process and $Z = g(S_T)$ for $g\colon \mathbb{R}^d \to \mathbb{R}$ and with simplistic market frictions we may know that the optimal strategy in (2) is of the simpler form $\delta_k = f_k(I_k, \delta_{k-1})$ for some $f_k\colon \mathbb{R}^{r+d} \to \mathbb{R}^d$.

Remark 7 We would similarly transform (4) into a modified unconstrained problem, optimized over $\mathcal{H}_M$.

Remark 8 For practical implementations, handling trading constraints with 10 is not particularly efficient since the gradient of Θ_M of our objective outside $\mathcal{H}$ vanishes. In the case where $H \circ \delta = \delta$ for $\delta \in \mathcal{H}$, this can be addressed by variants of

$$\pi(X) \equiv \inf_{\delta\in\mathcal{H}^u} \rho\,(X + (H\circ\delta\cdot S)_T - C_T(\delta) - \gamma\|\delta \\ -H\circ\delta\|_1)\,.$$

for Lagrange multipliers $\gamma \gg 0$.

The next proposition shows that thanks to the universal approximation theorem, strategies in $\mathcal{H}$ are approximated arbitrarily well by strategies in $\mathcal{H}_M$. Consequently, the neural network price $\pi^M(-Z) - \pi^M(0)$ converges to the exact price $p(Z)$.

Proposition 4.3 *Define $\mathcal{H}_M$ as in (9) and π^M as in (10). Then for any $X \in \mathcal{X}$,*

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

Proof We first note that the argument δ_{k-1} in 10 is redundant, since iteratively δ_{k-1} is itself a function of $I_0, \dots, I_{k-1}$. We may therefore write for the purpose of this proof

$$\mathcal{H}_M = \{(\delta_k^\theta)_{k=0,\dots,n-1} \in \mathcal{H}^u : \delta_k^\theta = F_k(I_0, \dots, I_k)\,, \\ F_k \in \mathcal{NN}_{M,r(k+1),d}\}. \tag{4.1'}$$

Since $\mathcal{H}_M \subset \mathcal{H}_{M+1} \subset \mathcal{H}^u$ for all $M \in \mathbb{N}$ it follows that $\pi^M(X) \geq \pi^{M+1}(X) \geq \pi(X)$. Thus it suffices to show that for any $\varepsilon > 0$ there exists $M \in \mathbb{N}$ such that $\pi^M(X) \leq \pi(X) + \varepsilon$.

By definition, there exists $\delta \in \mathcal{H}^u$ such that

$$\rho(X + (H\circ\delta\cdot S)_T - C_T(H\circ\delta)) \leq \pi(X) + \frac{\varepsilon}{2}. \tag{11}$$

Since δ_k is $\mathcal{F}_k$-measurable, there exists $f_k\colon \mathbb{R}^{r(k+1)} \to \mathbb{R}^d$ measurable such that $\delta_k = f_k(I_0, \dots, I_k)$ for each $k = 0, \dots, n-1$. Since Ω is finite, δ_k is bounded and so $f_k^i \in L^1(\mathbb{R}^{r(k+1)}, \mu)$ for any $i = 1, \dots d$, where μ is the law of $(I_0, \dots, I_k)$ under $\mathbb{P}$. Thus one may use Theorem 4.1 to find $F_{k,n}^i \in \mathcal{NN}_{\infty,r(k+1),1}$ such that $F_{k,n}^i(I_0, \dots, I_k)$ converges to $f_k^i(I_0, \dots, I_k)$ in $L^1(\mathbb{P})$ as $n \to \infty$.

By passing now to a suitable subsequence, convergence holds $\mathbb{P}$-a.s. simultaneously for all i,k. Writing $\delta_k^n := F_{k,n}(I_0, \dots, I_k)$ and using $\mathbb{P}[\{\omega\}] > 0$ for all $\omega \in \Omega$, this implies

$$\lim_{n\to\infty} \delta_k^n(\omega) = \delta_k(\omega) \quad \text{for all } \omega \in \Omega. \tag{12}$$

Continuity of $H_k(\cdot)(\omega)$ for a fixed ω implies moreover that also $\lim_{n\to\infty} H_k(\omega) \circ \delta_k^n(\omega) = H_k(\omega) \circ \delta_k(\omega)$.

Since Ω is finite, ρ can be viewed as a convex function $\rho\colon \mathbb{R}^N \to \mathbb{R}$. In particular, ρ is continuous. Using continuity of ρ in the first step and upper semi-continuity of $c_k(\cdot)(\omega)$ for each $\omega \in \Omega$ combined with monotonicity of ρ in the second step, one obtains

$$\begin{aligned} &\liminf_{n\to\infty} \rho(X + (H\circ\delta^n\cdot S)_T - C_T(H\circ\delta^n)) \\ &\quad\leq\ \rho(X + (H\circ\delta\cdot S)_T - \limsup_{n\to\infty} C_T(H\circ\delta^n)) \\ &\quad\leq\ \rho(X + (H\circ\delta\cdot S)_T - C_T(H\circ\delta)). \end{aligned}$$

Combining this with (11), there exists $n \in \mathbb{N}$ (large enough) such that

$$\rho(X + (H\circ\delta^n\cdot S)_T - C_T(H\circ\delta^n)) \leq \pi(X) + \varepsilon. \tag{13}$$

Since $\delta^n \in \mathcal{H}_M$ for all M large enough, one obtains $\pi^M(X) \leq \pi(X) + \varepsilon$ by (10) and (13), as desired. ■

4.3. Numerical solution for OCE-risk measures

While Theorem 4.1 and Proposition 4.3 give a theoretical justification for using hedging strategies built from neural networks, we now turn to computational considerations: how can we calculate a (close-to) optimal parameter $\theta \in \Theta_M$ for (10)?

To explain the key ideas we focus on the case when ρ is an OCE-risk measure (see (6)) and no trading constraints are present, the case of general risk measures is treated below.

Inserting the definition of ρ, see (6), into (10), the optimization problem can be rewritten as

$$\pi^M(-Z) = \inf_{\bar\theta\in\Theta_M} \inf_{w\in\mathbb{R}} \Big\{ w + \mathbb{E}[\ell(Z - (\delta^{\bar\theta}\cdot S)_T \\ + C_T(\delta^{\bar\theta}) - w)] \Big\} = \inf_{\theta\in\Theta} J(\theta),$$

where $\Theta = \mathbb{R} \times \Theta_M$ and for $\theta = (w, \bar\theta) \in \Theta$,

$$J(\theta) := w + \mathbb{E}[\ell(Z - (\delta^{\bar\theta}\cdot S)_T + C_T(\delta^{\bar\theta}) - w)]. \tag{14}$$

Generally, to find a local minimum of a differentiable function J, one may use a *gradient descent* algorithm: Starting with an initial guess $\theta^{(0)}$, one iteratively defines

$$\theta^{(j+1)} = \theta^{(j)} - \eta_j \nabla J_j(\theta^{(j)}), \tag{15}$$

for some (small) $\eta_j > 0, j \in \mathbb{N}$ and with $J_j = J$. Under suitable assumptions on J and the sequence $\{\eta_j\}_{j\in\mathbb{N}}$, $\theta^{(j)}$ converges to a local minimum of J as $j \to \infty$. Of course, the success and feasibility of this algorithm crucially depends on two points: Firstly, can one avoid finding a local minimum instead of a global one? Secondly, can ∇J be calculated efficiently?

One of the key insights of deep learning is that for cost functions J built based on neural networks both of these problems can be dealt with simultaneously by using a variant of *stochastic gradient descent* and the *(error) backpropagation* algorithm. What this means in our context is that in each step j the expectation in (14) (which is in fact a weighted sum over all elements of the finite but potentially very large sample space Ω) is replaced by an expectation over a randomly (uniformly) chosen subset of Ω of size $N_{\text{batch}} \ll N$, so that J_j used in the update (15) is now given as

$$J_j(\theta) = w + \sum_{m=1}^{N_{\text{batch}}} \ell(Z(\omega_m^{(j)}) - (\delta^{\bar{\theta}} \cdot S)_T(\omega_m^{(j)}) + C_T(\delta^{\bar{\theta}})(\omega_m^{(j)}) \\ - w)\frac{N}{N_{\text{batch}}}\mathbb{P}[\{\omega_m^{(j)}\}]$$

for some $\omega_1^{(j)}, \dots, \omega_{N_{\text{batch}}}^{(j)} \in \Omega$. This is the simplest form of the (minibatch) stochastic gradient algorithm. Not only does it make the gradient computation a lot more efficient (or possible at all, if N is large) but it also avoids getting stuck in local minima: even if $\theta^{(j)}$ arrives at a local minimum at some j, it moves on afterwards (due to the randomness in the gradient). In order to calculate the gradient of J_j for each of the terms in the sum, one may now rely on the compositional structure of neural networks. If ℓ, c and σ are sufficiently differentiable and the derivatives are available in closed form, then one may use the chain rule to calculate the gradient of $F^{\bar{\theta}_k}$ with respect to θ analytically and the same holds for the gradient of J_j. Furthermore, these analytical expressions can be evaluated very efficiently using the so called backpropagation algorithm (see subsequent section).

While this certainly answers the second question posed above (efficiency), the first one (local minima) is only partially resolved, as there is no general result guaranteeing convergence to the global minimum in a reasonable amount of time. However, it is common belief that for sufficiently large neural networks, it is possible to arrive at a sufficiently low value of the cost function in a reasonable amount of time, see Goodfellow *et al.* (2016, Chapter 8).

Finally, note that for the experiments in Section 5 below we have used Adam, a more refined version of the stochastic gradient algorithm, as introduced in Kingma and Ba (2015) and also discussed in Goodfellow *et al.* (2016, Chapter 8.5.3).

REMARK 9 In the experiments in Section 5 below, the functions ℓ, c and σ are continuous but have only piecewise continuous derivatives. Nevertheless, similar techniques can be applied.

REMARK 10 Numerically, trading constraints can be handled by introducing Lagrange-multipliers or by imposing infinite trading cost outside the allowed trading range. Certain types of constraints can also be dealt with by the choice of activation function: for example, no short-selling constraints can be enforced by choosing a non-negative activation function σ. A systematic numerical treatment will be left for future research.

4.4. *Certainty equivalent of exponential utility*

The entropic risk measure (5) is a special case of an OCE-risk measure, as explained in Example 2. However, when applying the methodology explained in Section 4.3, there is no need to minimize over w: we may directly insert (5) into (10) to write

$$\pi^M(-Z) = \frac{1}{\lambda}\log \inf_{\theta \in \Theta_M} J(\theta),$$

where

$$J(\theta) := \mathbb{E}[\exp(-\lambda[-Z + (\delta^\theta \cdot S)_T - C_T(\delta^\theta)])]. \tag{16}$$

A close-to-optimal $\theta \in \Theta_M$ can then be found numerically as above.

4.5. *Extension to general risk measures*

As explained in Section 4.3, for OCE-risk measures the optimal hedging problem (10) is amenable to deep learning optimization techniques (i.e. variants of stochastic gradient descent) via (14). The key ingredient for this is that the objective J satisfies

(ML1) the gradient of J decomposes into a sum over the samples, i.e. $\nabla_\theta J(\theta) = \sum_{m=1}^N \nabla_\theta J(\theta, \omega_m)$ and

(ML2) $\nabla_\theta J(\theta, \omega_m)$ can be calculated efficiently for each m, i.e. using backpropagation.

The goal of the present section is to show that for a general class of convex risk measures (including all coherent ones) one can approximate (2) by a minimax problem over neural networks and that the objective functional of this approximate problem also has these two key properties, making it amenable to deep learning optimization techniques.

Denote by $\mathcal{P}$ the set of probability measures on $(\Omega, \mathcal{F})$. The following result serves as a starting point:

THEOREM 4.4 Robust representation of convex risk measures *Suppose $\rho\colon \mathcal{X} \to \mathbb{R}$ is a convex risk measure. Then ρ can be written as*

$$\rho(X) = \max_{\mathbb{Q}\in\mathcal{P}} \left(\mathbb{E}_\mathbb{Q}[-X] - \alpha(\mathbb{Q})\right), \quad X \in \mathcal{X}, \tag{17}$$

where $\alpha(\mathbb{Q}) := \sup_{X\in\mathcal{X}}(\mathbb{E}_\mathbb{Q}[-X] - \rho(X))$.

Proof Since for Ω finite the set of probability measures $\mathcal{P}$ coincides with the set of finitely additive, normalized set functions (appearing in Föllmer and Schied 2016, Theorem 4.16), the present statement follows directly from the cited theorem and Föllmer and Schied (2016, Remark 4.17). ■

The function $\alpha\colon \mathcal{P} \to \mathbb{R}$ is called the (minimal) penalty function of the risk measure ρ.

Since Ω is finite, $\mathcal{P}$ can be identified with the standard $N-1$ simplex in $\mathbb{R}^N$ and so (17) is an optimization over $\mathbb{R}^N$. However, N is very large in our context and so the representation (17) is of little use for numerical calculations. The next result shows that $\rho(X)$ can be approximated by an optimization problem over a lower dimensional space. To state it, let us define the set $\mathcal{L} \subset \mathcal{X}$ of log-likelihoods by

$$\mathcal{L} := \{f \in \mathcal{X} : \mathbb{E}[\exp(f)] = 1\},$$

define $\bar{\alpha}\colon \mathcal{L} \to \mathbb{R}$ by $\bar{\alpha}(f) = \alpha(\exp(f)\mathrm{d}\mathbb{P})$ for any $f \in \mathcal{L}$ and write $\mathcal{P}_{eq}$ for the set of probability measures on $(\Omega, \mathcal{F})$, which are equivalent to $\mathbb{P}$. Furthermore, one may view $\bar{I} = (I_0, \dots, I_n)$ as a map $\Omega \to \mathbb{R}^{r(n+1)}$.

Theorem 4.5 *Suppose*

(i) $\alpha(\mathbb{Q}) < \infty$ *for some* $\mathbb{Q} \in \mathcal{P}_{eq}$,
(ii) $\bar{\alpha}$ *is continuous,*
(iii) $\mathcal{F} = \mathcal{F}_T$.

Then for any $X \in \mathcal{X}$, $\rho(X) = \lim_{M\to\infty} \rho^M(X)$, *where*

$$\rho^M(X) := \sup_{\substack{\theta\in\Theta_{M,r(n+1),1} \\ \mathbb{E}[\exp(F^\theta\circ\bar{I})]=1}} \left(\mathbb{E}[-X\exp(F^\theta\circ\bar{I})] - \bar{\alpha}(F^\theta\circ\bar{I})\right). \tag{18}$$

Proof We proceed in two steps. In a first step we show that for any $X \in \mathcal{X}$ one may write

$$\rho(X) = \sup_{\substack{\bar{f}\in\mathcal{M} \\ \mathbb{E}[\exp(\bar{f}\circ\bar{I})]=1}} \left(\mathbb{E}[-X\exp(\bar{f}\circ\bar{I})] - \bar{\alpha}(\bar{f}\circ\bar{I})\right), \tag{19}$$

where $\mathcal{M}$ denotes the set of measurable functions mapping from $\mathbb{R}^{r(n+1)} \to \mathbb{R}$. In the second step we rely on (19) to prove the statement.

Step 1: Since $\mathbb{P}[\{\omega_i\}] > 0$ for all i, $\mathcal{X}$ coincides with $L^\infty(\Omega, \mathcal{F}, \mathbb{P})$ and ρ is law-invariant. Thus by (i) and (Föllmer and Schied 2016, Theorem 4.43) one may write

$$\rho(X) = \sup_{\mathbb{Q}\in\mathcal{P}_{eq}} \left(\mathbb{E}_\mathbb{Q}[-X] - \alpha(\mathbb{Q})\right), \quad X \in \mathcal{X}. \tag{20}$$

Note that $\mathcal{P}_{eq}$ may be written in terms of $\mathcal{L}$ as

$$\mathcal{P}_{eq} = \{\exp(f)\mathrm{d}\mathbb{P} : f \in \mathcal{L}\}. \tag{21}$$

Furthermore, using (iii) one obtains

$$\mathcal{X} = \{\bar{f}\circ\bar{I} : \bar{f} \in \mathcal{M}\}. \tag{22}$$

Combining (20), (21) and the definition of $\bar{\alpha}$ one obtains

$$\rho(X) = \sup_{f\in\mathcal{L}} \left(\mathbb{E}[-X\exp(f)] - \bar{\alpha}(f)\right),$$

which can be rewritten as (19) by using (22).

Step 2: Note that one may also write (18) as

$$\rho^M(X) = \sup_{\substack{f\in\mathcal{NN}_{M,r(n+1),1} \\ \mathbb{E}[\exp(f\circ\bar{I})]=1}} \left(\mathbb{E}[-X\exp(f\circ\bar{I})] - \bar{\alpha}(f\circ\bar{I})\right). \tag{23}$$

Combining (23) with (19) and using $\mathcal{NN}_{M,r(n+1),1} \subset \mathcal{NN}_{M+1,r(n+1),1} \subset \mathcal{M}$, one obtains that $\rho^M(X) \leq \rho^{M+1}(X) \leq \rho(X)$ for all $M \in \mathbb{N}$. Thus it suffices to show that for any $\varepsilon > 0$ there exists $M \in \mathbb{N}$ such that $\rho^M(X) \geq \rho(X) - \varepsilon$.

By (19), for any $\varepsilon > 0$ one finds $\bar{f} \in \mathcal{M}$ such that

$$\mathbb{E}[\exp(\bar{f}\circ\bar{I})] = 1, \tag{24}$$

$$\rho(X) - \frac{\varepsilon}{2} \leq \mathbb{E}[-X\exp(\bar{f}\circ\bar{I})] - \bar{\alpha}(\bar{f}\circ\bar{I}). \tag{25}$$

Precisely as in the proof of Proposition 4.3, one may use Theorem 4.1 to find $f^{(n)} \in \mathcal{NN}_{\infty,r(n+1),1}$ such that $\mathbb{P}$-a.s., $f^{(n)}\circ\bar{I}$ converges to $\bar{f}\circ\bar{I}$ as $n \to \infty$. Combining this with (24), one obtains that for all n large enough, $c_n := \log(\mathbb{E}[\exp(f^{(n)}\circ\bar{I})])$ is well-defined and that $\bar{f}^{(n)}\circ\bar{I}$ also converges $\mathbb{P}$-a.s. to $\bar{f}\circ\bar{I}$, as $n \to \infty$, where $\bar{f}^{(n)} := f^{(n)} - c_n$. Using this, (25) and assumption (ii), for some (in fact all) $n \in \mathbb{N}$ large enough one obtains

$$\rho(X) - \varepsilon \leq \mathbb{E}[-X\exp(\bar{f}^{(n)}\circ\bar{I})] - \bar{\alpha}(\bar{f}^{(n)}\circ\bar{I}). \tag{26}$$

From $\mathcal{NN}_{\infty,r(n+1),1} - c_n = \mathcal{NN}_{\infty,r(n+1),1}$ and from the choice of $\mathcal{NN}_{M,r(n+1),1}$, one has $\bar{f}^{(n)} \in \mathcal{NN}_{M,r(n+1),1}$ for M large enough. By combining this with (26) and the choice of c_n one obtains

$$\rho(X) - \varepsilon \leq \rho^M(X),$$

as desired. ■

Combining (10) and (18), one thus approximates (2) for $X = -Z$ by solving

$$\inf_{\theta_0\in\Theta_M} \sup_{\theta_1\in\Theta_{M,r(n+1),1}} J(\theta), \tag{27}$$

where $\theta = (\theta_0, \theta_1)$,

$$J(\theta) := \mathbb{E}\left[-\mathrm{PL}(Z, 0, \delta^{\theta_0})\exp(F^{\theta_1}\circ\bar{I})\right] - \bar{\alpha}(F^{\theta_1}\circ\bar{I})$$
$$- \lambda_0(\mathbb{E}[\exp(F^{\theta_1}\circ\bar{I})] - 1)$$

and λ_0 is a Lagrange multiplier.

We conclude this section by arguing that the objective J in (27) indeed satisfies (ML1) and (ML2). This is standard (c.f. Section 4.3) for all terms in the sum except for $\bar{\alpha}(F^{\theta_1}\circ\bar{I})$ and so we only consider this term.

Recall that Ω is finite and consists of N elements, thus $\mathcal{X} = \{X : \Omega \to \mathbb{R}\}$ can be identified with $\mathbb{R}^N$. As for standard backpropagation the compositional structure can be used for efficient computation:

Proposition 4.6 *Suppose* $\bar{\alpha}$ *can be extended to* $\bar{\alpha} : \mathcal{X} \to \mathbb{R}$ *continuously differentiable,* σ *is continuously differentiable and* $\mathcal{NN}_{M,r(n+1),1}$ *is the set of neural networks with a fixed architecture (see Remark 4). Then* $J(\theta_1) := \bar{\alpha}(F^{\theta_1}\circ\bar{I})$, $\theta_1 \in \Theta_{M,r(n+1),1}$ *is continuously differentiable and satisfies (ML1).*

Proof Note that $F = F^{\theta_1}$ is parametrized by the matrices A^ℓ and vectors b^ℓ, $\ell = 1, \ldots, L$, and that one may consider all partial derivatives separately. Given $\bar{\alpha} : \mathcal{X} \to \mathbb{R}$ and $\nabla\bar{\alpha} : \mathcal{X} \to \mathcal{X}$, one thus aims at calculating $\partial_{A^\ell_{i,j}}\bar{\alpha}(F\circ\bar{I})$ and $\partial_{b^\ell_i}\bar{\alpha}(F\circ\bar{I})$ for $\ell = 1, \ldots, L, i = 1, \ldots, N_\ell, j = 1, \ldots, N_{\ell-1}$. This can be done by the chain rule: For $\theta \in \{A^\ell_{i,j}, b^\ell_i\}$, one has

$$\partial_\theta\bar{\alpha}(F\circ\bar{I}) = \sum_{m=1}^N \nabla\bar{\alpha}(F\circ\bar{I})(\omega_m)\partial_\theta F(\bar{I}(\omega_m))$$

and in particular (ML1) holds. ■

Furthermore, in the notation of the proof, for any $m = 1, \ldots N$ the derivative $\partial_\theta F(\bar{I}(\omega_m))$ can be calculated using standard backpropagation algorithm (preceded by a forward iteration) and so (ML2) holds as well. For the reader's convenience we state it here: One sets $x^0 = \bar{I}(\omega_m)$, iteratively calculates $x^\ell := F_\ell(x^{\ell-1})$ for $\ell = 1, \ldots, L-1$ and $x^L := W_L(x^{L-1})$.

Then (this is the backward pass) one sets $J^L := A^L$ and calculates iteratively $J^\ell = J^{\ell+1} dF_\ell(x^{\ell-1})$ for $\ell = L-1, \dots, 1$, where

$$dF_\ell(x^{\ell-1}) = \operatorname{diag}(\sigma'(W_\ell x^{\ell-1}))A^\ell.$$

From this one may use again the chain rule to obtain for any $\ell = 1, \dots L, i = 1, \dots, N_\ell, j = 1, \dots, N_{\ell-1}$ the derivatives of F with respect to the parameters as

$$\partial_{A^\ell_{i,j}} F(\bar{I}(\omega_m)) = J_i^{\ell+1} \sigma'((W_\ell x^{\ell-1})_i) x_j^{\ell-1}$$
$$\partial_{b^\ell_i} F(\bar{I}(\omega_m)) = J_i^{\ell+1} \sigma'((W_\ell x^{\ell-1})_i).$$

5. Numerical experiments and results

After having introduced the optimal hedging problem (2) in Section 3 and described in Section 4 how one may numerically approximate the solution by (10) using neural networks, we now turn to numerical experiments to illustrate the feasibility of the approach. We start by explaining in Section 5.1 the modeling choices in detail. The remainder of this section will then be devoted to examining the following questions:

- Section 5.2: How does neural network hedging (for different risk-preferences) compare to the benchmark in a Heston model without transaction costs?
- Section 5.3: What is the effect of proportional transaction costs on the exponential utility indifference price?
- Section 5.4: Is the numerical method scalable to higher dimensions?
- Section 5.5: How does it work in practice, e.g. when compared to a daily-recalibrated Black–Scholes model on the S&P500 index?

5.1. *Setting and implementation*

For most of the results presented here we have chosen a time horizon of 30 trading days with daily rebalancing. Thus, $T = 30/365$, $n = 30$ and the trading dates are $t_k = k/365$, $k = 0, \dots, n$. As explained in Section 4 and Remark 6, the number of units $\delta_k \in \mathbb{R}^d$ that the agent decides to hold in each of the instruments at t_k is parametrized by a semi-recurrent neural network: we set $\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$ where F^{θ_k} is a feed forward neural network with two hidden layers and $I_k = \Phi(S_0, \dots, S_k)$ for some $\Phi\colon \mathbb{R}^{(k+1)d} \to \mathbb{R}^d$ specified below. More precisely, in the notation of Definition 2, F^{θ_k} is a neural network with $L = 3$, $N_0 = 2d$, $N_1 = N_2 = d + 15$, $N_3 = d$ and the activation function is always chosen as $\sigma(x) = \max(x, 0)$. The weight matrices and biases are the parameters to be optimized in (10). Note that these are different for each k.

Having made these choices, the algorithm outlined in Section 4 can now be used for approximate hedging *in any market situation*: given sample trajectories of the hedging instruments $S(\omega_m)$, samples of the payoff $Z(\omega_m)$ and associated weights $\mathbb{P}[\{\omega_m\}]$ for $m = 1, \dots, N$ (on a finite probability space $\Omega = \{\omega_1, \dots, \omega_N\}$), for any choice of transaction cost structure c and any risk measure ρ one may now use the algorithm outlined in Section 4 to calculate close-to optimal hedging strategies and approximate minimal prices. Of course, for a path-dependent derivative with payoff $Z = G(S_0, \dots, S_T)$ with $G\colon (\mathbb{R}^d)^{n+1} \to \mathbb{R}$ one obtains samples of the payoff by simply evaluating G on the sample trajectories of S.

Different risk measures ρ, transaction cost functions c and payoffs Z will be used in the examples and so these are described separately in each of the subsequent sections. To illustrate the feasibility of the algorithm and have a benchmark at hand for comparison (at least in the absence of transaction costs), we have chosen to generate the sample paths of S from a standard stochastic volatility model under a risk-neutral measure $\mathbb{P}$ (except in Section 5.5). Thus in most of the examples below, the process S follows (a discretization of) a Heston model, see the beginning of Section 5.2 below. But we stress again that, as explained above, the algorithm is *model independent* in the sense that no information about the Heston model is used except for the (weighted) samples of the price and variance process.

The algorithm has been implemented in PYTHON, using TENSORFLOW to build and train the neural networks. To allow for a larger learning rate, the technique of batch normalization (see Ioffe and Szegedy 2015 and Goodfellow *et al.* 2016, Chapter 8.7.1) is used in each layer of each network right before applying the activation function. The network parameters are initialized randomly (drawn from uniform and normal distribution). For network training the Adam algorithm (see Kingma and Ba 2015, Goodfellow *et al.* 2016, Chapter 8.5.3) with a learning rate of 0.005 and a batch size of 256 has been used. Finally, the model hedge for the benchmark in Section 5.2 has been calculated using Quantlib.

REMARK 11 For most of the numerical experiments in this article the optimality criteria in (14) and (16) are specified under a risk-neutral measure. Thus, an optimal hedging strategy is based on market anticipations of future prices. Alternatively, one could use a statistical measure. The algorithm presented here can be applied also in this case, see Section 5.5.

REMARK 12 In the examples presented here the network architecture (number of layers and number of neurons in each layer) has been fixed. In future applications, one may include an automatic hyperparameter optimization procedure that selects the best network architecture, learning rate and batch size within a pre-specified set of choices, see e.g. Goodfellow *et al.* (2016, Chapter 11.4).

REMARK 13 Note that $\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$, i.e. the neural network at t_k takes as an input the output δ_{k-1}^θ of the network at t_{k-1}. By iterating one obtains that the strategy at t_k can also be viewed as a feed forward neural network with inputs $(I_0, \dots, I_k)$ and Lk hidden layers. Thus, even in case $L = 3$ one works with a 'deep' network.

5.2. *Benchmark: no transaction costs*

As a first example, we consider hedging without transaction costs in a Heston model. In this example the risk measure ρ is chosen as the average value at risk (also called conditional value at risk or expected shortfall), defined for any random variable X by

$$\rho(X) := \frac{1}{1-\alpha}\int_0^{1-\alpha} \mathrm{VaR}_\gamma(X)\mathrm{d}\gamma \qquad (28)$$

for some $\alpha \in [0,1)$, where $\mathrm{VaR}_\gamma(X) := \inf\{m \in \mathbb{R} : \mathbb{P}(X < -m) \leq \gamma\}$. An alternative representation of ρ of type (6) is discussed in Example 3. We refer to (Föllmer and Schied 2016, Section 4.4) for further details. Note that different levels of α correspond to different levels of risk-aversion, ranging from risk-neutral for α close to 0 to very risk-averse for α close to 1. The limiting cases are $\rho(X) = -\mathbb{E}[X]$ for $\alpha = 0$ and $\lim_{\alpha\uparrow 1}\rho(X) = -\mathrm{essinf}(X)$, see (Föllmer and Schied 2016, p. 234 and Remark 4.50).

5.2.1. A brief reminder on the Heston model.

Recall that a Heston model is specified by the stochastic differential equations

$$\begin{aligned} \mathrm{d}S^1_t &= \sqrt{V_t}S^1_t\mathrm{d}B_t, \quad \text{for } t > 0 \quad \text{and} \quad S^1_0 = s_0 \\ \mathrm{d}V_t &= \alpha(b - V_t)\mathrm{d}t + \sigma\sqrt{V_t}\mathrm{d}W_t, \quad \text{for } t > 0 \quad \text{and} \\ V_0 &= v_0, \end{aligned} \qquad (29)$$

where B and W are one-dimensional Brownian motions (under a probability measure $\mathbb{Q}$) with correlation $\rho \in [-1,1]$ and α, b, σ, v_0 and s_0 are positive constants. Below we have chosen $\alpha = 1$, $b = 0.04$, $\rho = -0.7$, $\sigma = 2$, $v_0 = 0.04$ and $s_0 = 100$, reflecting a typical situation in an equity market.

Here S^1 is the price of a liquidly tradeable asset and V is the (stochastic) variance process of S^1, modeled by a Cox–Ingersoll–Ross (CIR) process. V itself is not tradable directly but only through options on variance. In our framework this is modeled by an idealized variance swap with maturity T, i.e. we set $\mathcal{F}^H_t := \sigma((S^1_s, V_s) : s \in [0,t])$ and

$$S^2_t := \mathbb{E}_{\mathbb{Q}}\left[\int_0^T V_s\,\mathrm{d}s\middle|\mathcal{F}^H_t\right], \quad t \in [0,T], \qquad (30)$$

and consider (S^1, S^2) as the prices of liquidly tradeable assets. A standard calculation[†] shows that (30) is given as

$$S^2_t = \int_0^t V_s\,\mathrm{d}s + L(t, V_t) \qquad (31)$$

where

$$L(t,v) = \frac{v-b}{\alpha}(1 - \mathrm{e}^{-\alpha(T-t)}) + b(T-t).$$

Consider now a European option with payoff $g(S^1_T)$ at T for some $g\colon \mathbb{R} \to \mathbb{R}$. Its price (under $\mathbb{Q}$) at $t \in [0,T]$ is given as $H_t := \mathbb{E}_{\mathbb{Q}}[g(S^1_T)|\mathcal{F}^H_t]$. By the Markov property of (S^1, V), one may write the option price at t as $H_t = u(t, S^1_t, V_t)$ for some $u\colon [0,T] \times [0,\infty)^2 \to \mathbb{R}$. Assuming that u is sufficiently smooth, one may apply Itô's formula to H and use (31) to obtain

$$g(S^1_T) = q + \int_0^T \delta^1_t\,\mathrm{d}S^1_t + \int_0^T \delta^2_t\,\mathrm{d}S^2_t, \qquad (32)$$

where $q = \mathbb{E}_{\mathbb{Q}}[g(S^1_T)]$ and

$$\delta^1_t := \partial_s u(t, S^1_t, V_t)\text{and}\delta^2_t := \frac{\partial_v u(t, S^1_t, V_t)}{\partial_v L(t, V_t)}. \qquad (33)$$

Thus, if continuous-time trading was possible, (32) shows that the option payoff can be replicated perfectly by trading in (S^1, S^2) according to the strategy (33).

Remark 14 The strategy (33) depends on V_t. Although not observable directly, an estimate can be obtained by estimating $\int_0^t V_s\,\mathrm{d}s$ and solving (31) for V_t.

5.2.2. Setting: discretized Heston model.

In addition to the setting explained in detail in Section 5.1, here we set $d = 2$, consider no transaction costs (i.e. $C_T \equiv 0$) and generate sample trajectories of the price process of the hedging instruments from a discretely sampled Heston model. Thus, $S = (S_0, \ldots, S_n)$ and for any $k = 0, \ldots, n$, $S_k = (S^1_k, S^2_k)$ is given by (29) and (31) under $\mathbb{Q}$. The sample paths of S are generated by (exact) sampling from the transition density of the CIR process (see Glasserman 2004, Section 3.4) and then using the (simplified) Brodie–Kaya scheme (see Broadie and Kaya 2006; Andersen *et al.* 2010).[‡] Generating independent samples of S according to this scheme can now be viewed as sampling from a uniform distribution on a (huge) finite probability space Ω.[§] Thus, in the notation of Section 5.1 one has $\mathbb{P}[\{\omega_m\}] = 1/N$ for all $m = 1, \ldots, N$ with each $S(\omega_m)$ corresponding to a sample of the Heston model generated as explained above.

If continuous-time trading was possible, any European option could be replicated perfectly by following the strategy (33). However, in the present setup the hedging portfolio can only be adjusted at discrete time-points. Nevertheless one may choose $\delta^H_k := (\delta^1_k, \delta^2_k)$ for $k = 0 \ldots n-1$ with δ^1, δ^2 defined by (33) and charge the risk-neutral price q. This will be referred to as the model-delta hedging strategy (or simply model hedge) and serves as a benchmark.

Finally, in order to compare the neural network strategies to this benchmark, the network input is chosen as $I_k = (\log(S^1_k), V_k)$. One could also replace V_k by S^2_k instead. The network structure at time-step t_k is illustrated in figure 1.

5.2.3. Results.

We now compare the model hedge δ^H to the deep hedging strategies δ^θ corresponding to different risk-preferences, captured by different levels of α in the average value at risk (28).

[†] For example, one may use that $(\log(S^1), V)$ is an affine process to see that the conditional expectation in (30) can be taken only with respect to $\sigma(V_t, s \in [0,t])$. This conditional expectation can then be calculated by using the SDE for V or by directly inserting the expression from e.g. (Dufresne 2001, Section 3).

[‡] This corresponds to replacing V in the SDE for S^1 in (29) by a piecewise constant process and the integral in (31) by a sum.

[§] To be more precise, one replaces the normal distributions appearing in the simulation scheme for S by (arbitrarily fine) discrete distributions.

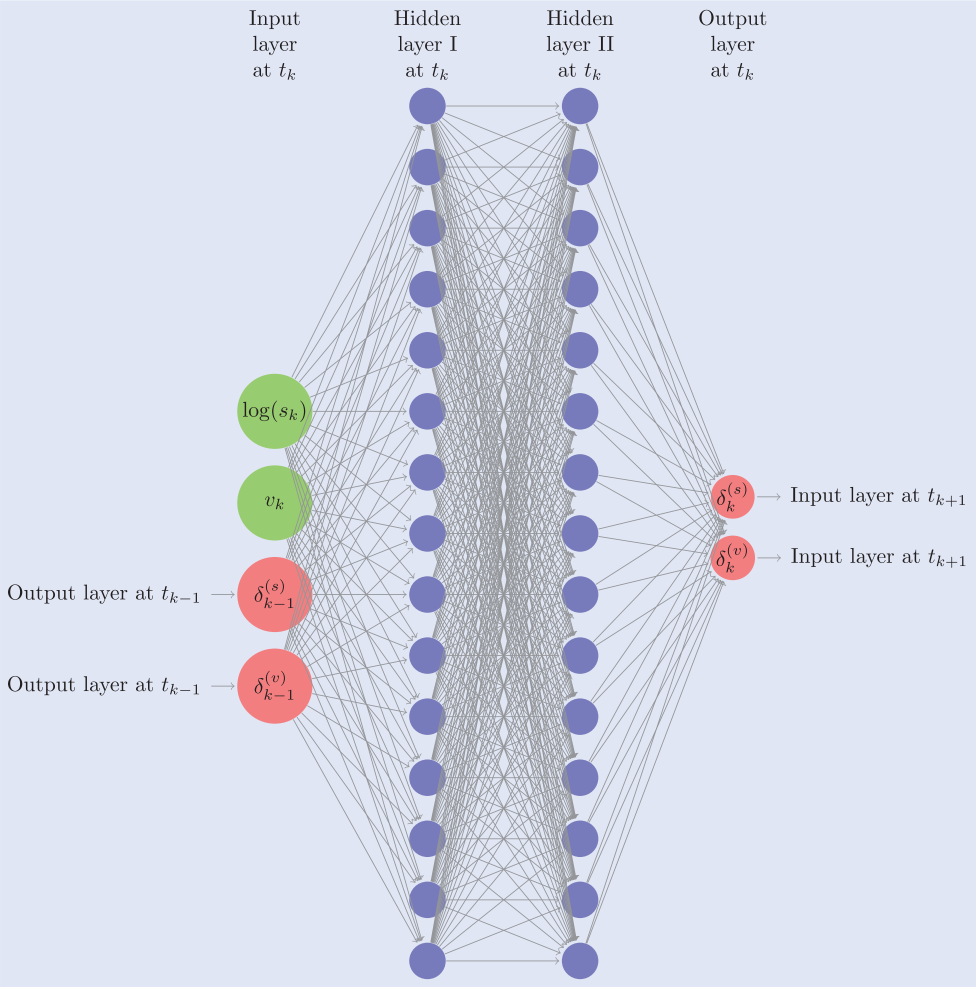


Figure 1. Visualization of the structure of the network strategy δ^θ at t_k. Input variables to δ^θ_k are the current values of I (green nodes on the left) and the previous holdings in the hedging instruments (red nodes on the left). The network passes these through two hidden layers (blue nodes in the middle) and returns the current hedging position δ^θ_k (red nodes on the right). The latter is then used as an input to δ^θ_{k+1}, the strategy at t_{k+1}.

As a first example, consider a European call option, i.e. $Z = (S^1_T - K)^+$ with $K = s_0$. Following the methodology outlined in Section 5.1, we calculate a (close-to) optimal parameter θ for (10) with $X = -Z$ and denote by δ^θ and p^θ_0 the (close-to) optimal hedging strategy and value of (10), respectively. By definition of the indifference price (3), the approximation property Proposition 4.3, Proposition 3.6 and $\rho(0) = 0$, p^θ_0 is an approximation to the indifference price $p(Z)$. As an out-of-sample test, one can then simulate another set of sample trajectories (here 10^6) and evaluate the terminal hedging errors $q - Z + (\delta^H \cdot S)_T$ (model hedge) and $p^\theta_0 - Z + (\delta^\theta \cdot S)_T$ (CVar) on each of them. In fact, since the risk-adjusted price p^θ_0 is higher than the risk-neutral price $q=1.69$ (as shown in Proposition 3.6(ii), e.g. for $\alpha = 0.5$ one obtains $p^\theta_0 = 1.94$), for (CVar) we have evaluated $q - Z + (\delta^\theta \cdot S)_T$, i.e. the hedging error from using the optimal strategy associated to ρ but only charging the risk-neutral price q. The terminal hedging errors associated to the two strategies are compared in a histogram in figure 2, where α in (28) is chosen as $\alpha = 0.5$. As one can see, the hedging performance of δ^H and δ^θ is very similar. In particular

- for this choice of risk-preferences (ρ as in (28) with $\alpha = 0.5$) the optimal strategy in (2) is close to the model hedge δ^H,
- the neural network strategy δ^θ is able to approximate well the optimal strategy in (2).

This is also illustrated by figure 3, where the first components of the strategies δ^θ_k and δ^H_k at a fixed time point k are plotted conditional on $(S^1_k, V_k) = (s, v)$ on a grid of values for (s, v). To make this last comparison fully sensible instead of the recurrent network structure $\delta^\theta_k = F^{\theta_k}(I_k, \delta^\theta_{k-1})$ here a simpler structure $\delta^\theta_k = F^{\theta_k}(I_k)$ is used. The hedging performance for this simpler structure is, however, very similar, see figure 4, where the terminal hedging errors associated to the two strategies are compared in a histogram. Of course, this is also expected from (33).†

† For non-zero transaction costs this is not true anymore, i.e. the recurrent network structure is needed. For example, figure 5 is generated for precisely the same parameters as figure 4, except that $\alpha = 0.99$ and proportional transaction costs are incurred, i.e. (34) with $\varepsilon = 0.01$. The histogram and the table in figure 5 show that the recurrent network structure realizes a better hedging performance at a lower price.

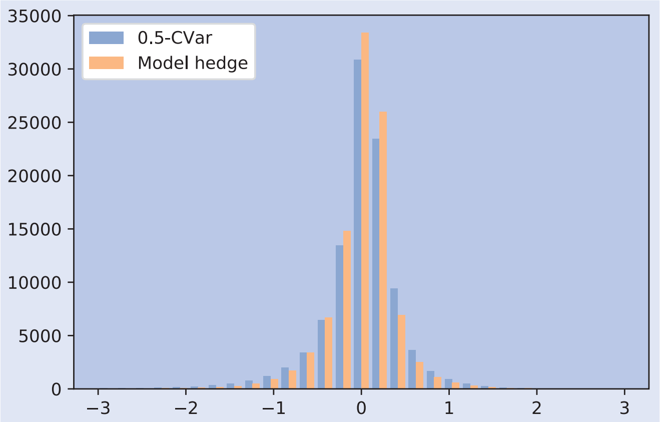


Figure 2. Comparison of model hedge and deep hedge associated to the 50%-expected shortfall criterion. The figure shows a histogram of the terminal hedging error $q - Z + (\delta \cdot S)_T$ evaluated on the test data, for both $\delta = \delta^H$ (model hedge) and $\delta = \delta^\theta$ (neural network hedge). The latter was optimized with $\alpha = 0.5$ in (28).

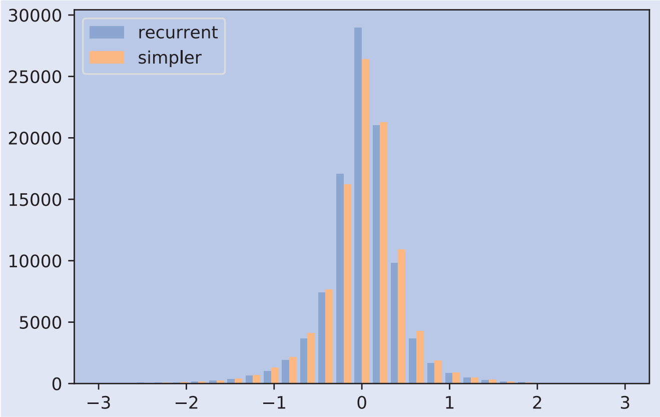


Figure 4. Comparison of deep hedging strategies with recurrent structure to those with a simpler network structure when no transaction costs are incurred. The figure shows a histogram of the terminal hedging error $q - Z + (\delta^\theta \cdot S)_T$ evaluated on the test data, for both $\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$ (recurrent) and $\delta_k^\theta = F^{\theta_k}(I_k)$ (simpler). Both strategies were optimized with $\alpha = 0.5$ in (28).

A more extreme case is shown in figure 6, where the terminal hedging errors associated to the strategies optimized for the 50%-CVar and 99%-CVar criterion (i.e. $\alpha = 0.5$ and $\alpha = 0.99$) are compared in a histogram. Also note that for $\alpha = 0.99$ one obtains a significantly higher risk-adjusted price $p_0^\theta = 3.49$. The histogram in figure 7 is the same as in figure 6, except that now the terminal hedging errors for both strategies are centered to the risk-neutral price q, i.e. one trades according to the 50% and 99%-CVar optimal hedging strategies but only charges the risk-neutral price. From this comparison the risk preferences become apparent, see the histogram in figure 7: the 50%-CVar strategy is more centered at 0 and also has a smaller mean hedging error but the 99%-expected shortfall strategy yields smaller extreme losses (c.f. also the realized 99%-CVar *loss* value realized on the test sample, shown in the table in figure 7).

To further illustrate the implications of risk-preferences on hedging, as a last example we consider selling a call spread, i.e. $Z = [(S_T^1 - K_1)^+ - (S_T^1 - K_2)^+]/(K_2 - K_1)$ for $K_1 < K_2$. Here we have chosen $K_1 = s_0$, $K_2 = 101$. Proceeding as above, we compare the model hedge to the more risk-averse hedging strategies associated to $\alpha = 0.95$ and $\alpha = 0.99$. The first components of the strategies (on a grid of values for spot and variance) are shown at a fixed time point in figures 8

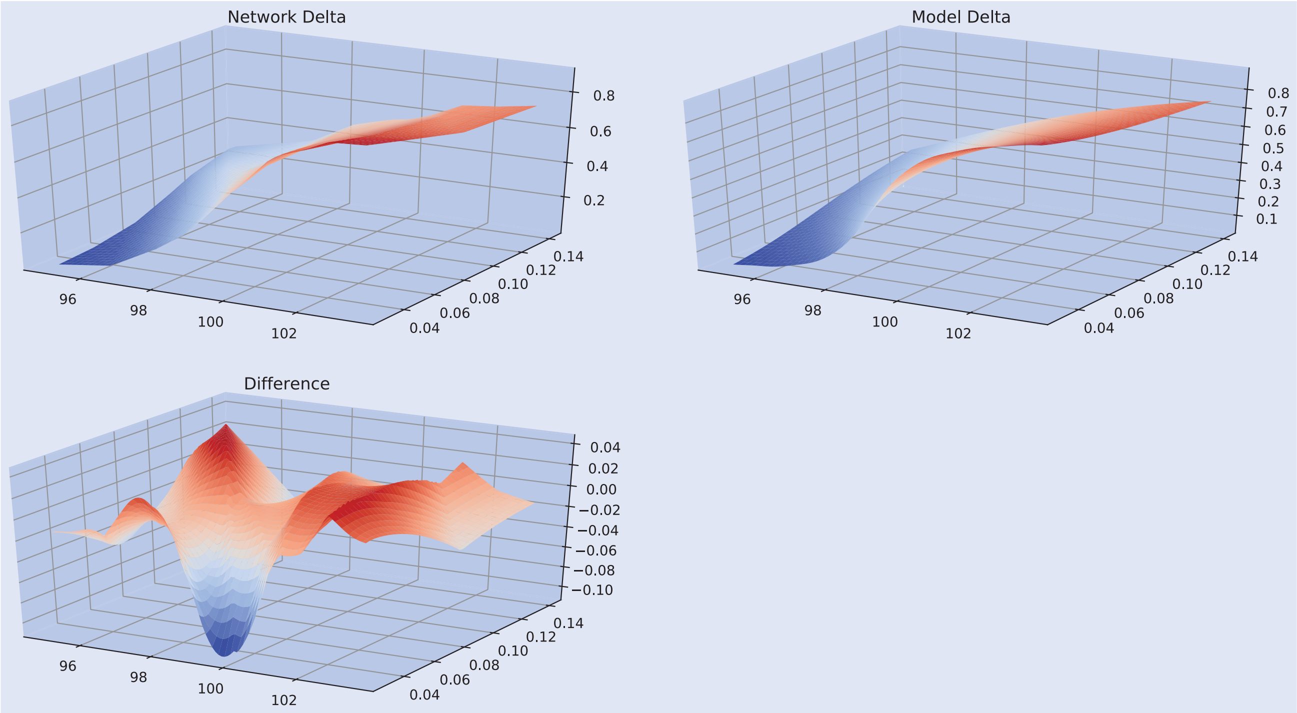
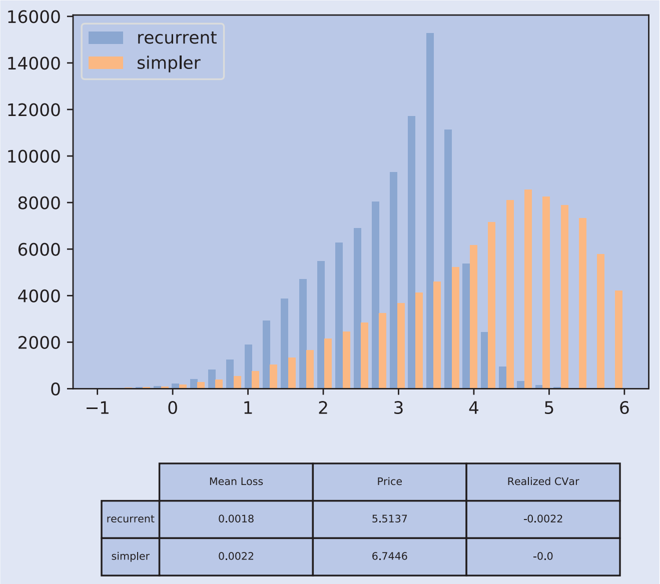


Figure 3. Comparison of holdings in stock for model hedge and neural network hedge at $k = 15$ days. The figure shows the first component of both δ_k^H (model delta) and δ_k^θ (network delta) as well as their difference plotted as a function of $I_k = (S_k, V_k)$, evaluated on a grid containing $[95, 103] \times [0.3, 0.13]$. The neural network hedge has the simpler structure $\delta_k^\theta = F^{\theta_k}(I_k)$ and was optimized using $\alpha = 0.5$ in (28).



	Mean Loss	Price	Realized CVar
recurrent	0.0018	5.5137	-0.0022
simpler	0.0022	6.7446	-0.0

Figure 5. Comparison of deep hedging strategies with recurrent structure to those with a simpler network structure in the presence of proportional transaction costs ((34) with $\varepsilon = 0.01$). The figure shows a histogram of the terminal hedging error $p_0^\theta - Z + (\delta^\theta \cdot S)_T - C_T(\delta^\theta)$ evaluated on the test data, for both $\delta_k^\theta = F^{\theta_k}(I_k, \delta_{k-1}^\theta)$ (recurrent) and $\delta_k^\theta = F^{\theta_k}(I_k)$ (simpler). Both strategies were optimized with $\alpha = 0.99$ in (28).

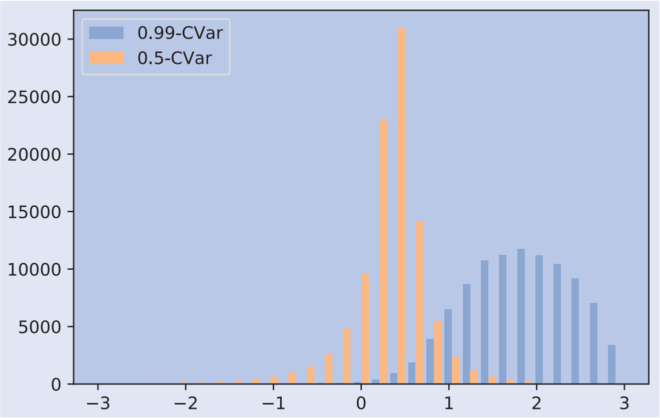
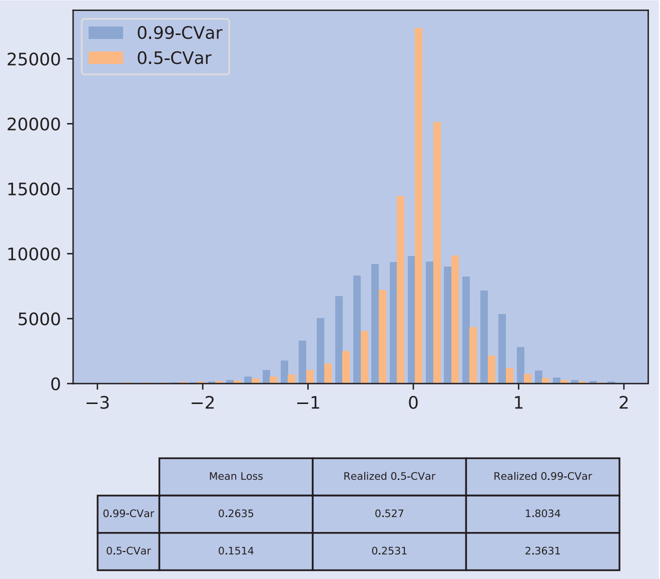


Figure 6. Comparison of deep hedging strategies for two different levels of risk-aversion. The figure shows a histogram of the terminal hedging error $p_0^\theta - Z + (\delta^\theta \cdot S)_T$ evaluated on the test data for two different choices of network strategy δ^θ. In both cases the risk measure (28) was used to optimize the strategy but once the parameter α in (28) was chosen as $\alpha = 0.99$ (0.99-CVar) and once it was chosen as $\alpha = 0.5$ (0.5-CVar).

and 9. The model hedge would again correspond to $\alpha = 0.5$. As one can see for higher levels of risk-aversion, the strategy flattens. From a practical perspective, this precisely corresponds to a barrier shift, i.e. a more risk-averse hedge for a call spread with strikes K_1 and K_2 actually aims at hedging a spread with strikes $\tilde{K}_1$ and K_2 for $\tilde{K}_1 < K_1$.

5.3. *Price asymptotics under proportional transaction costs*

In Section 5.2 we have seen that in a market without transaction costs, deep hedging is able to recover the model hedge



	Mean Loss	Realized 0.5-CVar	Realized 0.99-CVar
0.99-CVar	0.2635	0.527	1.8034
0.5-CVar	0.1514	0.2531	2.3631

Figure 7. Comparison of deep hedging strategies for two different levels of risk-aversion, centered at the risk-neutral price. The figure shows a histogram of the terminal hedging error $q - Z + (\delta^\theta \cdot S)_T$ evaluated on the test data for two different choices of network strategy δ^θ. In both cases the risk measure (28) was used to optimize the strategy but once the parameter α in (28) was chosen as $\alpha = 0.99$ (0.99-CVar) and once it was chosen as $\alpha = 0.5$ (0.5-CVar).

and can be used to calculate risk-adjusted optimal hedging strategies.

The goal of this section is to illustrate the power of the methodology by numerically calculating the indifference price (3) in a multi-asset market with transaction costs.

So far, this has been regarded a highly challenging problem, see e.g. the introduction of Kallsen and Muhle-Karbe (2015). For example, calculating the exponential utility indifference price for a call option in a Black–Scholes model involves solving a multidimensional non-linear free boundary problem, see e.g. Hodges and Neuberger (1989) and Davis *et al.* (1993). Motivated by this Whalley and Wilmott (1997) have studied asymptotically optimal strategies and price asymptotics for small proportional transaction costs, i.e. for

$$c_k(n) = \sum_{i=1}^{d} \varepsilon |n^i| S_k^i \tag{34}$$

and as $\varepsilon \downarrow 0$. One of the results in the asymptotic analysis is that

$$p_\varepsilon - p_0 = O(\varepsilon^{2/3}), \quad \text{as } \varepsilon \downarrow 0, \tag{35}$$

where $p_\varepsilon = p_\varepsilon(Z)$ is the utility indifference price of Z associated to transaction costs of size ε. In fact (35) is true in more general one-dimensional models, see Kallsen and Muhle-Karbe (2015), and the rate 2/3 also emerges in a variety of related problems with proportional transaction costs, see e.g. Rogers (2004) and Muhle-Karbe *et al.* (2017) and the references therein.

Here we numerically verify (35) using the deep hedging algorithm, first for a Black–Scholes model (for which (35) is known to hold) and then for a Heston model (with $d = 2$ hedging instruments). For this latter case (or any other model

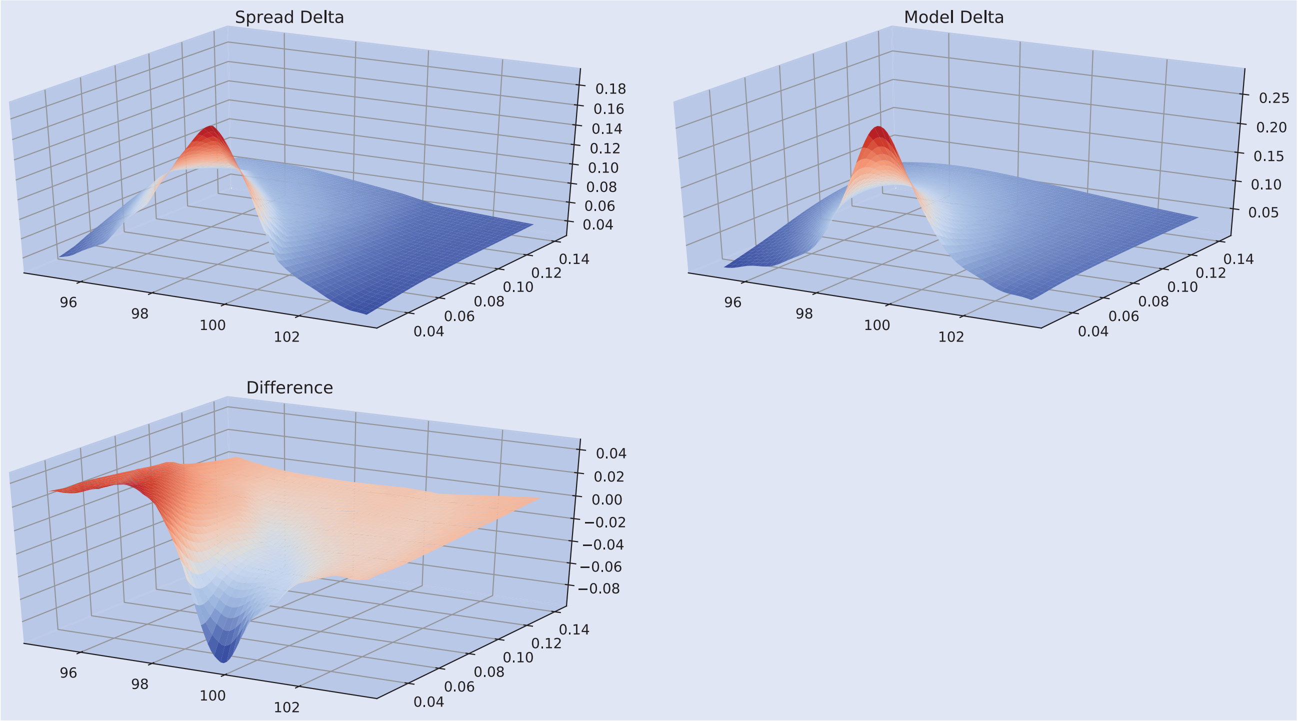


Figure 8. Comparison of holdings in stock for model hedge and 95%-CVar neural network hedge at $k = 15$ days when hedging a call spread. The figure shows the first component of both δ_k^H (model delta) and δ_k^θ (network delta) as well as their difference plotted as a function of $I_k = (S_k, V_k)$, evaluated on a grid containing $[95, 103] \times [0.3, 0.13]$. The neural network hedge has the simpler structure $\delta_k^\theta = F^{\theta_k}(I_k)$ and was optimized using $\alpha = 0.95$ in (28).

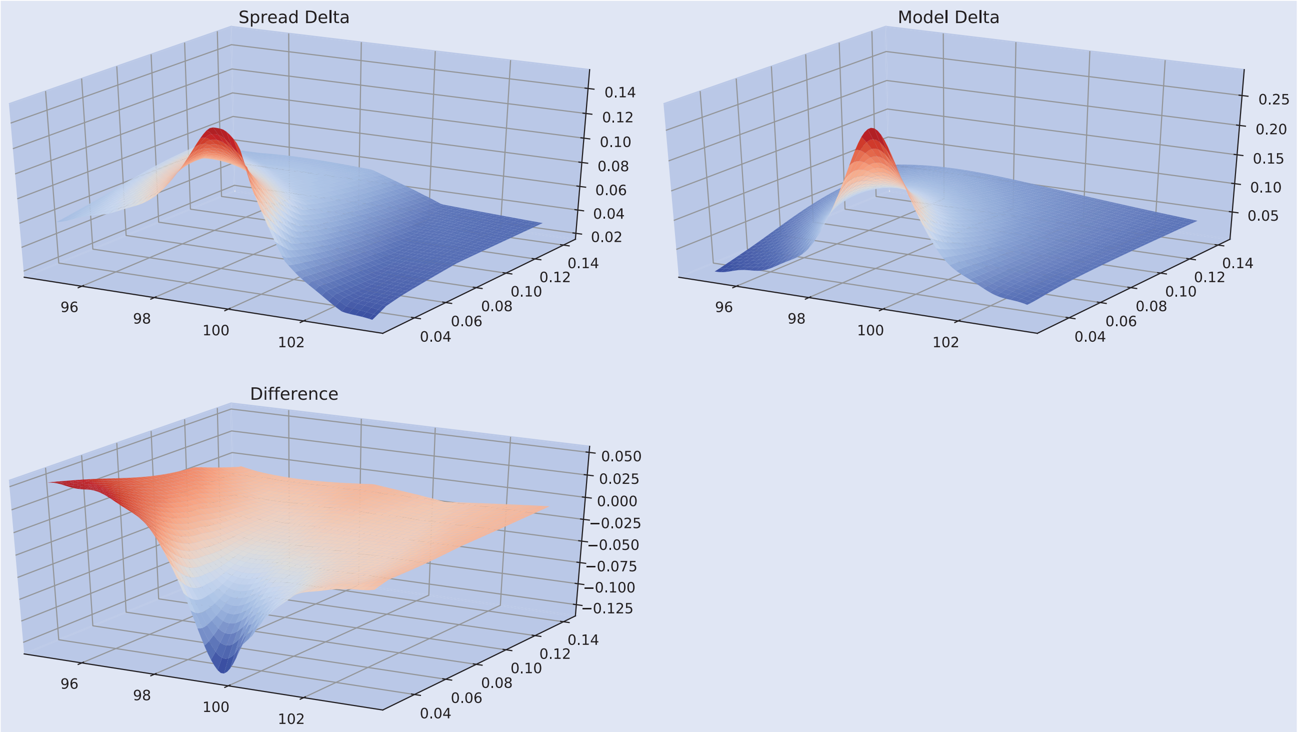


Figure 9. Comparison of holdings in stock for model hedge and 99%-CVar neural network hedge at $k = 15$ days when hedging a call spread. The figure shows the first component of both δ_k^H (model delta) and δ_k^θ (network delta) as well as their difference plotted as a function of $I_k = (S_k, V_k)$, evaluated on a grid containing $[95, 103] \times [0.3, 0.13]$. The neural network hedge has the simpler structure $\delta_k^\theta = F^{\theta_k}(I_k)$ and was optimized using $\alpha = 0.99$ in (28).

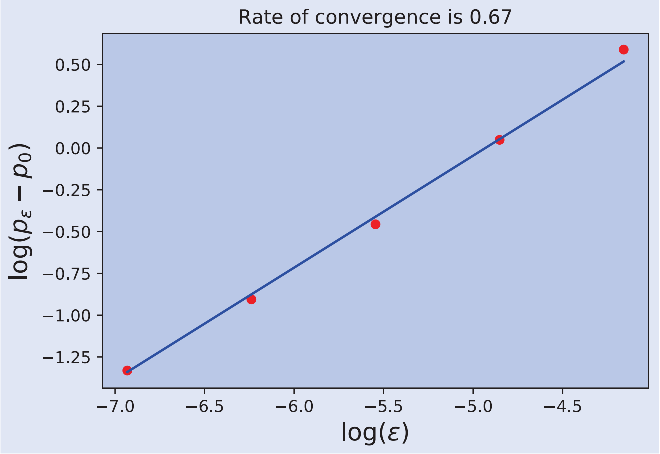


Figure 10. Black–Scholes model price asymptotics for proportional transaction costs. The figure shows the pairs $(\log(\varepsilon_i), \log(p_{\varepsilon_i} - p_0))$ (in red) and the closest (in squared distance) line with slope 2/3 (in blue), where p_{ε_i} is the neural network approximation to the exponential utility indifference price in a Black–Scholes model with proportional transaction costs (34) of size $\varepsilon_i = 2^{-i+5}$, $i = 1, \ldots, 5$.

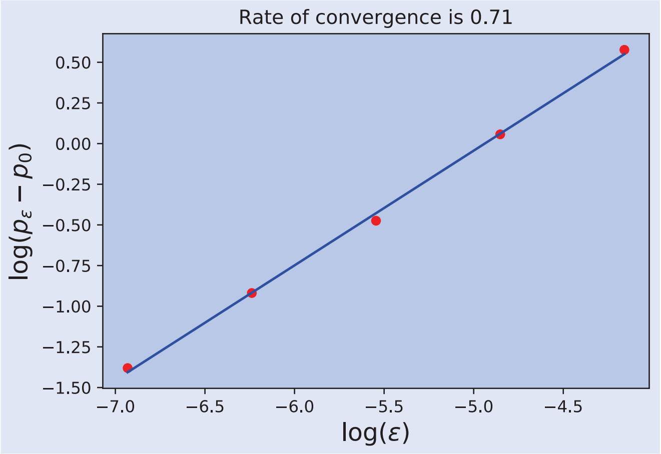


Figure 11. Heston model price asymptotics for proportional transaction costs. The figure shows the pairs $(\log(\varepsilon_i), \log(p_{\varepsilon_i} - p_0))$ (in red) and the regression line (in blue), where p_{ε_i} is the neural network approximation to the exponential utility indifference price in a Heston model with proportional transaction costs (34) of size $\varepsilon_i = 2^{-i+5}$, $i = 1, \ldots, 5$.

with $d > 1$) there have been neither numerical nor theoretical results on (35) previously in the literature.

5.3.1. Black–Scholes model. Consider first $d = 1$ and $S_t = s_0 \exp(-t\sigma^2/2 + \sigma W_t)$, where $\sigma > 0$ and W is a one-dimensional Brownian motion. We choose $\sigma = 0.2$, $s_0 = 100$ and use the explicit form of S to generate sample trajectories. Setting $I_k = \log(S_k)$ and proceeding precisely as in the Heston case (see Sections 5.1 and 5.2), we may use the deep hedging algorithm to calculate the exponential utility indifference price p_ε for different values of ε. Recall that we choose proportional transaction costs (34) and ρ is the entropic risk measure (5) (see Lemma 3.4). For the numerical example we take $\lambda = 1$ and $Z = (S_T - K)^+$ with $K = s_0$ and we calculate p_{ε_i} for $\varepsilon_i = 2^{-i+5}$, $i = 1, \ldots, 5$.

Figure 10 shows the pairs $(\log(\varepsilon_i), \log(p_{\varepsilon_i} - p_0))$ (in red) and the closest (in squared distance) straight line with slope 2/3 (in blue). Thus, in this range of ε the relation $\log(p_\varepsilon - p_0) = 2/3 \log(\varepsilon) + C$ for some $C \in \mathbb{R}$ indeed holds true and hence also (35).

Note that trading is only possible at discrete time-points and so the indifference price and the risk-neutral price do not coincide. Since (35) is a result for continuous-time trading (where $q = p_0$), we have compared to the risk-neutral price q here (thus neglecting the discrete-time friction in p_ε for $\varepsilon > 0$).

5.3.2. Heston model. We now consider a Heston model with two hedging instruments, i.e. $d = 2$ and the setting is precisely as in Section 5.2, except that here ρ is chosen as (5) and proportional transaction costs (34) are incurred. Choosing $\lambda = 1$, $Z = (S_T^1 - K)^+$ and ε_i as in the Black–Scholes case above, one can again calculate the exponential utility indifference prices and show the difference to p_0 in a log-log plot (see above) in a graph. These are shown as red dots in figure 11. Here the blue line in figure 11 is the regression line, i.e. the least squares fit of the red dots. The rate is very close to 2/3 and so it appears that the relation (35) also holds in this case.

5.4. *High-dimensional example*

As a last example consider a model built from 5 separate Heston models, i.e. $d = 10$ and (S^h, S^{h+1}) is the price process of spot and variance swap in a Heston model (specified by (29) and (31)) for $h = 1, \ldots, 5$. To have a benchmark at hand the 5 models are assumed independent and each of them has parameters as specified in Section 5.2. This choice is of course no restriction for the algorithm and is only made for convenience. The payoff is a sum of call options on each of the underlyings, i.e. $Z = \sum_{h=1}^{5} Z_h$ with $Z_h = (S_T^{2h-1} - K)^+$ and $K = s_0 = 100$. In a market with continuous-time trading and no transaction costs, Z can be replicated perfectly by trading according to strategy (33) in each of the models. In particular, this strategy is decoupled, i.e. the optimal holdings in (S^h, S^{h+1}) only depend on $(S^{(h)}, S^{(h+1)})$. While in the present setup trading is only possible at discrete time-steps and so the strategy optimizing (2), where $X = -Z$, leads to a non-deterministic terminal hedging error (1), by independence one still expects that the optimal strategy is decoupled as above, at least for certain classes of risk measures. To see this most prominently, here we consider *variance optimal hedging*: the objective is chosen as (4) for $\ell(x) = x^2$ and $p_0 = 5q$, where $q = \mathbb{E}[Z_1]$.

Let $\delta \in \mathcal{H}$ and write $\delta^{(2h-1:2h)} := (\delta^{2h-1}, \delta^{2h})$ for $h = 1, \ldots, 5$ (and analogously for S). If δ is decoupled, i.e. such that $\delta^{(2h-1:2h)}$ is independent of $S^{(2j-1:2j)}$ for $j \neq h$, then by independence and since S is a martingale one has

$$\begin{aligned}
&\mathbb{E}\left[(-Z + p_0 + (\delta \cdot S)_T)^2\right] \\
&\quad = \sum_{h=1}^{5} \operatorname{Var}\left(-Z_i + (\delta^{(2h-1:2h)} \cdot S^{(2h-1:2h)})_T\right).
\end{aligned} \tag{36}$$

By building δ from the (discrete-time) variance optimal strategies for each of the five models, one sees from (36) that the minimal value of (4) over *all* $\delta \in \mathcal{H}$ is at most five times the minimal value of (4) associated to a single Heston model. This consideration serves as a guideline for assessing the approximation quality of the neural network strategy.

To assess the scalability of the algorithm, we now calculate the close-to-optimal neural network hedging strategy associated to (4) in both instances (i.e. for $n_H = 5$ models and for a single one, $n_H = 1$) and compare the results. Unless specified otherwise, the parameters are as in Section 5.1. Since for $n_H = 5$ we are actually solving five problems at once, we allow for a network with more hidden nodes by taking $N_1 = N_2 = 12n_H$. We then train both networks for a fixed number of iterations (here 2×10^5) and measure the performance in terms of both training time and realized loss (evaluated on a test set of $n_H \times 10^5$ sample paths): the training times on a standard Lenovo X1 Carbon laptop are 5.75 and 2.1 hours for $n_H = 5$ and $n_H = 1$, respectively and the realized losses are 1.13 and 0.20. In view of the considerations above, this indicates that the approximation quality is roughly the same for both instances (and close-to-optimal).

While far from a systematic study, this last example nevertheless demonstrates the potential of the algorithm for high-dimensional hedging problems.

5.5. *Hedging a call option on the S&P500 index*

The previous examples have shown that the deep hedging methodology can be used to calculate hedging strategies in complex models incorporating e.g. transaction costs. We now complement these results by assessing its performance in a simple practical situation, in which the (daily recalibrated) Black–Scholes hedging strategy is known to work very well. The example that we consider here is the problem of hedging an at-the-money European call option on the S&P500 index with time-to-maturity $T_{\text{days}} \in \{30, 240\}$. Let us now explain the experiment in detail. We perform it for different time windows and start by describing it for one of those.

Firstly, we download historical daily (adjusted) returns of the S&P500 index from 5 March 2013 to 2 January 2018. These $n_{\text{ret}} = 1217$ data-points are then split into a training period and a test period, i.e. the first $n_{\text{ret}} - T_{\text{days}}$ data-points are considered as the past (and present) and may be used for training the deep hedging strategy. The last T_{days} returns are considered as the future and will only be used as a test set for comparing the deep hedging strategy to the (daily recalibrated) Black–Scholes hedging strategy.

As a next step we train the deep hedging strategy based on the historical returns of the index in the training period. To do so, we pick a commonly used, well-established econometric model, fit it to returns in the training period and then generate 2×10^7 sample paths from it. Given these sample trajectories, we proceed precisely as in the previous examples to find a close-to-optimal neural network hedging strategy. This demonstrates again the model-free character of our approach. More specifically, we choose a GJR-GARCH model with t-distributed innovations, i.e. $\{z_k\}$ are i.i.d. t-distributed with ν degrees of freedom and

$$\begin{aligned} \log(S_k/S_{k-1}) &= \mu + \varepsilon_k, \\ \varepsilon_k &= \sigma_k z_k, \\ \sigma_k^2 &= \omega + \alpha\varepsilon_{k-1}^2 + \beta\sigma_{k-1}^2 + \gamma\varepsilon_{k-1}^2 1_{\{\varepsilon_{k-1}<0\}}, \end{aligned} \tag{37}$$

where $k \in \{-n_{\text{ret}} + T_{\text{days}}, \ldots, T_{\text{days}}\}$ so that $k = 0$ corresponds to the last point in the training period (‘today’). Given the data in the training period the parameters $\alpha, \beta, \gamma, \omega, \mu, \nu$ are estimated by using a maximum likelihood method and this gives in particular an estimate for σ_0. Initializing with that and the historical values of the S&P500 index for ε_0 and S_0 one may then generate $N = 2 \times 10^7$ sample paths of $S = (S_0, \ldots, S_{T_{\text{days}}})$ by simulating according to (37) for $k = 1, \ldots, T_{\text{days}}$ with the estimated parameters. These sample paths can then be used to train the neural network hedging strategy precisely as in the previous examples, see e.g. Section 5.1. Note that here $d = 1$, $I_k = \log(S_k)$, $T = T_{\text{days}}$, $Z = (S_T - K)^+$, $K = S_0$, the learning rate is set to 0.0001, $N_1 = N_2 = 20$, and, as commonly done, we only account for transaction costs by allowing less frequent rebalancing of the hedging strategy (weekly for $T_{\text{days}} = 30$ and biweekly for $T_{\text{days}} = 240$).†

Having trained the deep hedging strategy, one may now evaluate its performance on a test set of simulated paths and, more importantly, the testing period of the historical values of the S&P500 index. On each of these paths, one may also evaluate the performance of a practitioner’s Black–Scholes strategy, in which at every trading date the volatility plugged into the Black–Scholes strategy is extracted from observed prices (implied volatility, if available or filtered volatility). Here volatility is estimated using the filter induced by (37). Finally, to allow for a fair comparison, the terminal P&L associated to the deep hedging strategy is based on charging the Black–Scholes price instead of the indifference price $p(Z)$. Otherwise the deep hedge would provide even better results by charging higher premia.

As a first example we consider $T_{\text{days}} = 240$ and optimize the deep hedging strategy according to the objective (4) for $\ell(x) = x^2$ and $p_0 = q$, where q is the Black–Scholes price associated to the volatility σ_0. Figure 12 compares the terminal hedging errors for the two strategies in a histogram. The Black–Scholes price is at around 70.5 and, at the same price, the deep hedging strategy achieves a mean squared hedging loss which is around 20% lower. Note that these are simulated trajectories; however, if one believes that the chosen market simulator (37) is a realistic model, then the deep hedging methodology indeed yields an impressive increase in performance.

As a second example we consider $T_{\text{days}} = 30$ and choose ρ as (28) with $\alpha = 0.75$. We now repeat the experiment described above for a range of 300 different *time windows*, i.e. for every new experiment we shift the time window described above into the past by one week. Thus, e.g. the last experiment is precisely as described above except that the historical daily returns of the S&P500 index that are used for parameter estimation, training and testing range from 12 June 2007 to 10 April 2012. This procedure yields 300 observations of hedging errors for both strategies on historical data. Figure 13 compares the terminal hedging errors for the two strategies in a histogram. The mean terminal P&L for the Black–Scholes strategy is 3.2, whereas for the deep hedging strategy it is 4.3. The mean square hedging *loss* incurred by

† Therefore, e.g. for $T_{\text{days}} = 240$ the trading dates are $t_k = 10k$, $k = 0, \ldots, 24$, and the sample paths of prices used for hedging are obtained by picking from the generated samples of $(S_0, \ldots, S_{T_{\text{days}}})$ only the prices at these dates.

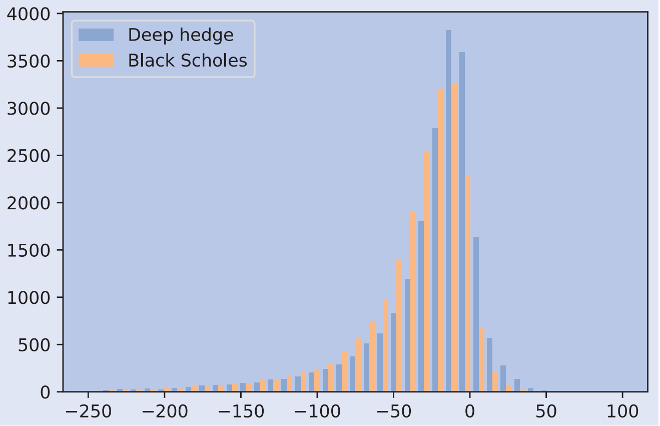


Figure 12. Comparison of (daily recalibrated) Black–Scholes model and deep hedge centered at the Black–Scholes price. The figure shows a histogram of the terminal hedging error $q - Z + (\delta \cdot S)_T$ evaluated on the simulated test data.

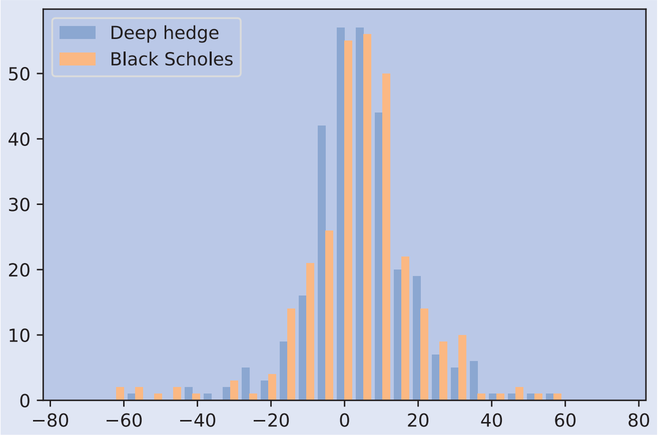


Figure 13. Comparison of (daily recalibrated) Black–Scholes model and deep hedge centered at the Black–Scholes price. The figure shows a histogram of the terminal hedging error $q - Z + (\delta \cdot S)_T$ evaluated on historical test data of the S&P500 index.

the Black–Scholes model is −11.6, whereas the deep hedging methodology yields a value of −7.9, i.e. larger losses are incurred in the Black–Scholes model on average.

Thus, this provides examples of settings where a refined Black–Scholes model is known (and empirically observed) to work well but the deep hedging methodology outperforms substantially. These examples also show that the methodology can be used to base hedging decisions on the *statistical measure*, which further illustrates the generic nature of the deep hedging approach.

6. Conclusion

This article considers the problem of hedging a portfolio of derivatives in a generic market environment with all sorts of frictions. Given a set of instruments available for hedging, the holdings in each of these instruments at any trading date are obtained by feeding (a suitable selection of) current (and possibly past) market information as well as the holdings of the previous trading date into a neural network. Mathematically, this corresponds to parametrizing hedging strategies by neural networks, which by well known universality results allows for ϵ optimal strategies. Given a market scenario generator (based on historical data or Monte Carlo samples of a classical derivatives model), the parameters of the network can be trained using state-of-the-art machine learning optimization techniques in the spirit of reinforcement learning. While in a frictionless market one would aim e.g. at minimizing the squared loss of the terminal hedging error, here the optimality criterion is specified by a convex risk measure, such as, for example, the average value at risk (expected shortfall). This framework allows to naturally incorporate market frictions as e.g. transaction costs and liquidity restrictions.

The article both lays the theoretical foundations for the approach and demonstrates its power in numerical experiments. On the one hand it provides a proof that the approximation of the 'minimal price' (the indifference price) provided by the neural network hedging strategy converges to the theoretical optimum as the complexity of the network increases. For a certain class of risk measures, namely optimized certainty equivalents, the close-to-optimal neural network hedge can be directly approximated using machine learning optimization techniques. For more general risk measures this is not the case but we show how an additional approximation based on the dual representation of the risk measure makes the problem amenable to backpropagation and stochastic gradient type algorithms.

On the other hand we present numerical experiments performed in PYTHON using TENSORFLOW. In most experiments we consider data generated from a Heston model and use both stock and a variance swap as hedging instruments. First we consider a market without transaction costs, for which the well known model hedge is available as a benchmark. We show that our methodology is able to accurately learn this benchmark strategy from samples, if the set of sample scenarios is rich enough. Furthermore, different levels of risk aversion allow to obtain hedging strategies which put more emphasis on avoiding extreme losses or minimizing the average hedging loss. We then demonstrate how our methodology can be used to numerically study the impact of proportional transaction costs on option prices. In particular, we calculate the asymptotic rate of convergence in a Heston model, which so far has only been known in one-dimensional models. We then present experiments for multiple Heston models, which show that the approach is feasible also in a high-dimensional setting. Finally, we complement these results by an experiment on historical data of the S&P500 index, in which we compare the methodology to a (daily recalibrated) Black–Scholes model and obtain a better hedging performance.

7. Disclaimer

Opinions and estimates constitute our judgement as of the date of this Material, are for informational purposes only and are subject to change without notice. This Material is not the product of J.P. Morgans Research Department and therefore, has not been prepared in accordance with legal requirements to promote the independence of research, including but not limited to, the prohibition on the dealing ahead of the dissemination of investment research. This Material is not intended as research, a recommendation, advice, offer or solicitation for the purchase or sale of any financial product or service, or to be used in any way for evaluating the merits of participating in any transaction. It is not a research report and is not intended as such. Past performance is not indicative of future results. Please consult your own advisors regarding legal, tax, accounting or any other aspects including suitability implications for your particular circumstances. J.P. Morgan disclaims any responsibility or liability whatsoever for the quality, accuracy or completeness of the information herein, and for any reliance on, or use of this material in any way.

Important disclosures at: www.jpmorgan.com/disclosures

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Alexander, C. and Nogueira, L.M., Model-free hedge ratios and scale-invariant models. *J. Bank. Finance*, 2007, **31**, 1839–1861.

Andersen, L.B.G., Jäckel, P. and Kahl, C., Simulation of square-root processes. In *Encyclopedia of Quantitative Finance*, 2010 (John Wiley and Sons).

Bank, P., Soner, H.M. and Voß, M., Hedging with temporary price impact. *Math. Financ. Econ.*, 2017, **11**, 215–239.

Barles, G. and Soner, H.M., Option pricing with transaction costs and a nonlinear Black-Scholes equation. *Finance Stoch.*, 1998, **2**, 369–397.

Bates, D.S., Hedging the smirk. *Finance Res. Lett.*, 2005, **2**, 195–200.

Ben-Tal, A. and Teboulle, M., An old-new concept of convex risk measures: The optimized certainty equivalent. *Math. Finance*, 2007, **17**, 449–476.

Bölcskei, H., Grohs, P., Kutyniok, G. and Petersen, P., Optimal approximation with sparsely connected deep neural networks. *Preprint arXiv:1705.01714*, 2017.

Bouchard, B., Moreau, L. and Soner, H.M., Hedging under an expected loss constraint with small transaction costs. *SIAM J. Financ. Math.*, 2016, **7**, 508–551.

Broadie, M. and Kaya, O., Exact simulation of Stochastic volatility and other affine jump diffusion processes. *Oper. Res.*, 2006, **54**, 217–231.

Burgert, C. and Rüschendorf, L., Consistent risk measures for portfolio vectors. *Insurance: Math. Econ.*, 2006, **38**, 289–297.

Carmona, R., *Indifference Pricing: Theory and Applications*, 2009 (Princeton University Press: Princeton, NJ).

Crépey, S., Delta-hedging vega risk? *Quant. Finance*, 2004, **4**, 559–579.

Davis, M.H.A., Panas, V.G. and Zariphopoulou, T., European option pricing with transaction costs. *SIAM J. Control Optim.*, 1993, **31**, 470–493.

Du, X., Zhai, J. and Lv, K., Algorithm trading using q-learning and recurrent reinforcement learning. *arxiv*, 2009. Available online at: https://arxiv.org/pdf/1707.07338.pdf.

Dufresne, D., The integrated square-root process, Centre for Actuarial Studies, Research Paper no. 90, University of Melbourne, 2001.

Dupire, B., Pricing with a smile. *Risk*, 1994, **7**, 18–20.

Föllmer, H. and Leukert, P., Efficient hedging: Cost versus shortfall risk. *Finance Stoch.*, 2000, **4**, 117–146.

Föllmer, H. and Schied, A., *Stochastic Finance: An Introduction in Discrete Time*, 2016 (De Gruyter: Berlin, Germany).

Gatheral, J. and Schied, A., Dynamical models of market impact and algorithms for order execution. In *Handbook on Systemic Risk*, pp. 579–599, 2013.

Glasserman, P., *Monte Carlo Methods in Financial Engineering, Applications of Mathematics: Stochastic Modelling and Applied Probability*, 2004 (Springer: New York).

Goodfellow, I., Bengio, Y. and Courville, A., *Deep Learning*, 2016 (MIT Press). Available online at: http://www.deeplearningbook.org.

Halperin, I., QLBS: Q-learner in the Black–Scholes (-Merton) Worlds. *arxiv*, 2017. Available online at: https://arxiv.org/abs/1712.04609.

Hodges, S. and Neuberger, A., Optimal replication of contingent claims under transaction costs. *Rev. Futures Markets*, 1989, **8**, 222–239.

Hornik, K., Approximation capabilities of multilayer feedforward networks. *Neural. Netw.*, 1991, **4**, 251–257.

Hull, J. and White, A., Optimal delta hedging for options. *J. Bank. Finance*, 2017, **82**, 180–190.

Hutchinson, J.M., Lo, A.W. and Poggio, T., A Nonparametric approach to pricing and hedging derivative securities via learning networks. *J. Finance.*, 1994, **49**, 851–889.

İlhan, A., Jonsson, M. and Sircar, R., Optimal static-dynamic hedges for exotic options under convex risk measures. *Stoch. Process. Their. Appl.*, 2009, **119**, 3608–3632.

Ioffe, S. and Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456, 2015.

Jiang, Z., Xu, D. and Liang, J., A deep reinforcement learning framework for the financial portfolio management problem. *arxiv*, 2017. Available online at: https://arxiv.org/abs/1706.10059.

Kabanov, Y. and Safarian, M., *Markets with Transaction Costs: Mathematical Theory*, 2009 (Springer: Berlin, Germany).

Kallsen, J. and Muhle-Karbe, J., Option pricing and hedging with small transaction costs. *Math. Finance*, 2015, **25**, 702–723.

Kingma, D.P. and Ba, J., Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Klöppel, S. and Schweizer, M., Dynamic indifference valuation via convex risk measures. *Math. Finance*, 2007, **17**, 599–627.

Lu, D., Agent inspired trading using recurrent reinforcement learning and LSTM neural networks. *arxiv*, 2017. Available online at: https://arxiv.org/pdf/1707.07338.pdf.

Moody, J. and Wu, L., Optimization of trading systems and portfolios. *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr)*, 1997, pp. 300–307.

Muhle-Karbe, J., Reppen, M. and Soner, H.M., A primer on portfolio choice with small transaction costs. *Annu. Rev. Financ. Econ.*, 2017, **9**, 301–331.

Rogers, L.C.G., Why is the effect of proportional transaction costs $O(\delta^{2/3})$. In *Mathematics of Finance*, edited by G. Yin and Q. Zhang, pp. 303–308, 2004 (American Mathematical Society: Providence, RI).

Rogers, L.C.G. and Singh, S., The cost of illiquidity and its effects on hedging. *Math. Finance*, 2010, **20**, 597–615.

Sepp, A., An approximate distribution of delta-hedging errors in a jump-diffusion model with discrete trading and transaction costs. *Quant. Finance*, 2012, **12**, 1119–1141.

Shaham, U., Cloninger, A. and Coifman, R.R., Provable approximation properties for deep neural networks. *Appl. Comput. Harmon. Anal.*, 2018, **44**, 537–557.

Soner, H.M., Shreve, S.E. and Cvitanić, J., There is no nontrivial hedging portfolio for option pricing with transaction costs. *Ann. Appl. Probab.*, 1995, **5**, 327–355.

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

Xu, M., Risk measure pricing and hedging in incomplete markets. *Ann. Finance*, 2006, **2**, 51–71.