Deep Hedging and Pricing for Noncontinuous Payoffs with Transaction Costs

September 25, 2020

Niclas Küpper
Master Thesis
ETH Zürich

Supervisors
Norberto Broggini
Josef Teichmann

Contents

1 Introduction

Pricing and hedging portfolios of derivatives is a decisive part of risk management in many industries. The Nobel Prize winning approach discovered by Robert C. Merton, Fischer Black, and Myron Scholes, now commonly called the Black-Scholes options pricing model, assumes a "complete market" where it is possible to set up a dynamic self-financing portfolio. This portfolio aims to replicate the derivative's payoff, thus eliminating any risk posed by the underlying asset. Then, since we assume an arbitrage free market, the "correct" price is given by the expected value of the option with respect to a martingale measure, also called risk-neutral measure, which is also precisely the price of implementing the replicating portfolio. This elegant solution is, however, not realistic.

In real markets trading is restricted by transaction costs, liquidity constraints, discrete trading opportunities, and market impacts. The Black-Scholes approach is not equipped to handle these restrictions. If we were to implement the Black-Scholes hedging portfolio (in continuous time) with transaction costs, we would incur an infinite cost in any trading interval no matter how small. Furthermore, the predicted price of the Black-Scholes model relies on calculating "greeks", which requires directly observable quantities like the spot price but also difficult to measure and generally non-constant variables such as the parameters of the underlying market model, whose assumptions might not be true.

To solve these problems, we turn to a deep learning approach. This fairly recent invention has dominated classical methods in many fields, from medical research to robotics. We choose this tool to generate trades that can address all of the above limitations. This approach only requires a market scenario generator, a risk measure, trading frictions/restrictions, and specified trading instruments.

Our focus will be the effect of transaction costs on the pricing and hedging of call and binary options. We will first recreate classical results in markets without transaction costs to demonstrate the capability of the deep approach on known problems. Then we move on to explore the effects of transaction costs and investigate how different models of networks perform in such market conditions. Finally we will look at convergence properties and other trade restrictions.

1.1 Related Works

Hedging and pricing in markets with frictions has been widely studied and many diverse approaches have been tried. Two more classical approaches include [11] and [4]. In the former, the classical delta hedge, that we derive in Section 3.4, is modified by adapting the volatility, making it dependent on the proportionality constant of the transaction costs. Specifically,

$$\hat{\sigma}^2(\sigma^2, k, \Delta t) = \sigma^2\big[1 + \frac{\sqrt{2/\pi}}{\sigma\sqrt{\Delta t}}k\big],$$

where σ is the original volatility, k is the original transaction cost, and Δt is the discretization time step. This method was proven to converge to replicate the option as the step size converges to zero, despite the transaction costs. In the latter a Hamilton-Jacobi-Bellman equation is derived from a utility maximization problem, which leads to a non-linear partial differential equation describing regions when to buy sell or hold. This pde is then solved using discrete time dynamic programming methods.

These ideas are expanded upon in [14] where the authors allow for a non-linear variable volatility term. Using utility maximization they not only uniquely identify the volatility term but also quantify the probability of missing the hedge in terms of the proportional transaction costs and the chosen utility function. Both [4] and [14] restrict themselves to European call options and exponential utility. We expand upon these ideas by allowing any kind of derivative and risk measure, including risk defined by utility functions.

Optimization under convex risk measures is also widely studied in other circumstances. In [8] a broader class of hedges was investigated where one is allowed to, in addition to trading the underlying, buy or sell vanilla options at the beginning of the trading period. So if, for a given convex risk measure ρ we would want to hedge an exotic claim G^e we would minimize

$$\rho\Big(- \lambda \cdot G - G^e + (\theta \cdot dS)_T + x + \lambda \cdot g\Big)$$

over (θ, λ) where θ is a trading strategy and λ is the number of vanilla options bought. Here G is the vector of outcomes of the vanilla options and g is a vector of the market prices for these options.

The paper [3] on deep hedging covers many similar results to this thesis. It shows also that, both theoretically and practically, neural networks can hedge various options under different convex risk measures. It also replicates a result of price convergence as proportional transaction costs converge as in [13]. Of course to build up the theoretical framework we need the well known result that deep feed forward networks satisfy universal approximation properties, see e.g. [7].

2 Setting: Discrete Time Market

The following setting is the same as in [3]. We consider a financial market in discrete time with a finite time horizon T with trading dates $0 = t_0 < \cdots <$

$t_n = T$. We fix a finite[1] probability space Ω with N elements. Market information is represented by $I = (I_k)_{k=0,\dots,n}$ where I_k represents the information at time t_k and takes values in $\mathbb{R}^r$. We then denote the filtration generated by I as $\mathcal{F} = (\mathcal{F}_k)_{k \leqslant n}$.

Our market then consists of d hedging instruments where the prices are given by an $\mathbb{R}^d$ valued $\mathcal{F}$-adapted stochastic process S. Note that it is *not* required that an equivalent martingale measure exists for S.

Our trading strategies are then given by an $\mathbb{R}^d$ valued $\mathcal{F}$-adapted stochastic process $\delta = (\delta_k)_{k \leq n}$, where $\delta_k = (\delta_k^i)_{i \leqslant d}$. Here δ_k^i represents the agents holdings in the i-th asset at the k-th time step.

We will consider binary[2] and European call options given respectively by evaluating

$$f(x) = 1_{\mathbb{R}_{\geq K}} \tag{2.1}$$

$$f(x) = \max(0, x - K) \tag{2.2}$$

on S_T, for some K. We call K the strike price. Although these options are not path dependent, this framework can also hedge path dependent options such as American options. See Appendix 7.5.

2.1 Hedging

Profits and losses when trading with strategy δ while being exposed to some risky asset Z are given by

$$\mathrm{PL}_T(Z, p, \delta) = -Z + p + (\delta \cdot S)_T - C_T(\delta), \tag{2.3}$$

where p is some initial cash injection (which is allowed to be negative) and C_t is a function describing the transaction costs incurred by following the trading strategy δ. The cumulative gains and losses of trading are described by:

$$(\delta \cdot S)_{t_k} = \sum_{j \leqslant k-1} \delta_j (S_{j+1} - S_j). \tag{2.4}$$

The cost of buying a position $\Delta \in \mathbb{R}^d$ in S at time t_k will be $c_k(\Delta)$. Then the total cost incurred is given by

$$C_T(\delta) = \sum_{k=1}^{n} c_k(\delta_k - \delta_{k-1}). \tag{2.5}$$

[1] We are considering numerical solutions, hence a finite probability space more accurately describes this setting.

[2] The author will freely switch between calling these options binary and digital. They are the same thing.

We assume that the function c_k is non-negative and upper semi-continuous. Further we assume that $c_k(0) = 0$, i.e. not trading costs nothing. Examples include

- Proportional transaction costs: for $c_k^i > 0$ define $c_k(\Delta) := \sum_{i \leqslant d} c_k^i S_k^i |\Delta^i|$.
- Fixed transaction costs: for $c_k^i > 0$ define $c_k(\Delta) := \sum_{i \leqslant d} c_k^i 1_{|\Delta^i| \geq \varepsilon}$.

Apart from the initial cash injection, all trading is self-financing, i.e. we run additionally a bank account with zero interest rate. If we call η_t the holdings in our bank account at time t, then at any given moment, for a given strategy δ, our total value is described by

$$V_t(\delta) = \sum_{i=1}^{d} \delta_t^i S_t^i + \eta_t - C_t(\delta),$$

i.e. the value of the stocks we own plus the money in our bank account minus incurred transaction costs. We then say that our trading is *self-financing* if $V_t(\delta) - ((\delta \cdot S)_t - C_t(\delta))$ is constant in t. This means that money moves into, or out of, the bank account only as a result of trading; profits and losses, and transaction costs. This uniquely defines the bank account η_t and so we will not explicitly mention it for the rest of the thesis.

This framework can be expanded to also consider trade restrictions, in which case we would replace δ_k with $H_k(\delta_k)$ everywhere for some $\mathcal{F}_k$-measurable restriction functions H_k. Restriction functions and their effect on trading are explored more in Section 5.4.

2.2 Convex Risk Measures

Since we are not necessarily dealing with a complete market and we have trading frictions, there does not necessarily exist a unique replication strategy for every liability Z. Thus we need to find some other optimality criterion to define an acceptable minimal price. We consider optimality under *convex risk measures* as studied in [5] and [8].

Let $\mathcal{X}$ be the set of random variables $X : \Omega \to \mathbb{R}$. Note that it is not necessary to assume that a specific probability measure is given on Ω, however we will consider a risk measure dependent on a probability measure for the rest of this thesis.

Definition 2.1. Consider we have asset positions $X_1, X_2 \in \mathcal{X}$. We call $\rho : \mathcal{X} \to \mathbb{R}$ a *convex risk measure* if it satisfies the following:

- Monotonically Decreasing: $X_1 \geq X_2$ a.s. $\implies \rho(X_1) \leqslant \rho(X_2)$.

- Convex: $\rho(\alpha X_1 + (1-\alpha)X_2) \leqslant \alpha\rho(X_1) + (1-\alpha)\rho(X_2)$, for $\alpha \in [0,1]$.
- Cash-invariant: $\rho(X + c) = \rho(X) - c$, for $c \in \mathbb{R}$.

We further call ρ *coherent* if it also satisfies:

- Positive Homogeneity: For $\lambda \geq 0$ we have $\rho(\lambda X) = \lambda\rho(X)$.

Lastly we say ρ is *normalized* if $\rho(0) = 0$.

The financial motivation behind monotonicity is clear: A strictly greater payoff poses a smaller risk. Convexity is motivated by weighing the avoidance of losses more highly than potential profits. Finally, cash-invariance comes from interpreting $\rho(X)$ as the needed cash to be indifferent to some position X. In particular, by cash-invariance we get

$$\rho(X + \rho(X)) = 0. \tag{2.6}$$

The above equation can be interpreted as saying that for a liability Z, $\rho(-Z)$ is a fair price since that is the amount of cash that needs to be added to the position to have zero risk.

Example 2.1. 1. The simplest example of a convex risk measure is negative expectation

$$\rho(X) = \mathbb{E}[-X], \quad X \in \mathcal{X},$$

it clearly satisfies all of the required conditions to be a coherent convex risk measure. It is, however, not a financially realistic measure, as it weighs profits and losses equally. It requires the choice of a probability measure.

2. A commonly used measure is the *entropic risk measure* given by

$$e_\gamma(X) = \frac{1}{\gamma}\log\big(\mathbb{E}[\exp(-\gamma X)]\big), \quad X \in \mathcal{X}, \gamma > 0.$$

It also requires the choice of a probability measure. The entropic risk measure is also a good example of an incoherent risk measure, this can easily be demonstrated with a Bernoulli random variable. It also exhibits a behaviour called constant absolute risk aversion, which implies, among other things, that optimal holding of an asset is independent of the level of initial wealth [1] [12]. We call γ the *risk aversion* parameter, it intuitively describes how risk-averse the agent is. Larger γ's describe a stronger avoidance of risk.

3. The *worst case measure* defined by

$$\rho_{\max}(X) = -\operatorname*{ess\,inf}_{\omega\in\Omega} X(\omega)$$

is an example of a convex risk measure independent of probability measures (up to equivalence). Note that in our setting we can replace the essential infimum by a minimum since we have a finite space. It is the most conservative measure in the sense that for any other (normalized) convex risk measure ρ we have that

$$\rho(X) \leqslant \rho(\operatorname*{ess\,inf}_{\omega \in \Omega} X(\omega)) = \rho_{\max}(X).$$

2.3 Optimal Hedging

Let us consider the following optimization problem for a given convex risk measure ρ and liability $Z \in \mathcal{X}$:

$$\pi(-Z) = \inf_{\delta \in \mathcal{H}} \rho(-Z + (\delta \cdot S)_T - C_T(\delta)). \tag{2.7}$$

where $\mathcal{H}$ is the family of $\mathbb{R}^d$-valued $\mathcal{F}$ adapted processes, i.e. the family of all possible trading strategies. We now define the indifference price $p(Z)$ as the price the trader needs to charge to be indifferent to the position $-Z$ when optimally trading with respect to the convex risk measure. In other words, there should be no difference between being exposed to $-Z$ having charged $p(Z)$ while trading all available assets and not doing anything. That is

$$\pi(-Z + p(Z)) = \pi(0). \tag{2.8}$$

We then calculate $p(Z)$ by iterating through possible prices and choosing whichever is closest to $\pi(0)$. In the case π is itself cash-invariant we can make calculating $p(Z)$ simpler by rewriting Equation 2.8 as

$$p(Z) = \pi(-Z) - \pi(0). \tag{2.9}$$

Then we only have to solve the optimization problem once.

We can see that without transaction costs, this price coincides with the price of the replicating portfolio (if it exists). The following lemma is taken from [3].

Lemma 2.2. Suppose $C_T \equiv 0$. For any $\delta^* \in \mathcal{H}$ and $p_0 \in \mathbb{R}$ we find that $p(Z) = p_0$ for $Z = p_0 + (\delta^* \cdot S)_T$.

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

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

Now taking the infimum over $\delta \in \mathcal{H}$ on both sides and using the fact that $\mathcal{H} - \delta^* = \mathcal{H}$ we find:

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

■

If we assume ρ to be normalized and S to be a continuous $\mathbb{P}-$martingale, for some probability measure $\mathbb{P}$, then $\pi(0) = 0$ for certain convex risk measures and the equation we want to solve becomes finding a price such that $\pi(-Z+p(Z)) = 0$. When we then write this out with the definition of π we get

$$\inf_{\delta\in\mathcal{H}} \rho(-Z + (\delta \cdot S)_T + p(Z) - C_T(\delta)) = 0. \tag{2.10}$$

Lemma 2.3. Let S be a continuous $\mathbb{P}-$martingale. Then, for all normalized convex risk measures of the form $\rho(X) = f^{-1}\mathbb{E}f(-X)$, with f convex, we have $\pi(0) = 0$.

Proof. First we show $\pi(0) \leqslant 0$. This follows directly from the fact that $\delta \equiv 0$ is a valid strategy, and so

$$\pi(0) = \inf_{\delta\in\mathcal{H}} \rho((\delta \cdot S)_T - C_T(\delta)) \leqslant \rho(0) = 0.$$

Next we show $\pi(0) \geqslant 0$. We use Jensen's inequality:

$$\pi(0) = \inf_{\delta\in\mathcal{H}} f^{-1}\mathbb{E}\big[f\big(- ((\delta \cdot S)_T - C_T(\delta))\big)\big] \geqslant \inf_{\delta\in\mathcal{H}} \mathbb{E}[-((\delta \cdot S)_T - C_T(\delta))] = 0.$$

■

In particular, the above lemma holds for the entropic risk measure, which is our main focus. This lemma also holds for shortfall and power risk, which can both be represented in the above way.

3 Neural Networks

3.1 Universal Approximation Theorem

The key idea continued from [3] is using neural networks to approximate hedging strategies. This works not only well due to the approximation properties of neural networks but also their numerical efficiency in finding feasible solutions. First, let us recall the definition of a neural network.

Definition 3.1 (Neural Network). Let $L, N_0, \ldots N_L \in \mathbb{N}$, $\sigma : \mathbb{R} \to \mathbb{R}$ and for all $l \in \{1, \ldots, L\}$ let $W_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ be an affine function. We then call a function $F : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ defined as

$$F(x) = W_L \circ G_{L-1} \circ \cdots \circ G_1, \quad \text{with } G_l = \sigma \circ W_l \tag{3.1}$$

a *(fully connected feed forward) neural network*. The *activation function* σ is applied componentwise. We call N_0 and N_L the input and output dimension, respectively, L the number of layers and $N_1, \dots, N_{L-1}$ the dimensions of the *hidden layers*.

We write the affine functions $W_l(x) = A^l x + b^l$ for $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and $b^l \in \mathbb{R}^{N_l}$. We call A^l the edge weights and b^l the biases.

Finally denote by $\mathcal{N}^{\sigma}_{\infty,d_0,d_1}$ the set of all deep neural networks with input dimension d_0 and output dimension d_1.

The next two results are versions of the Universal Approximation Theorem as in [7, Theorems 1 and 2].

Theorem 3.1. *If σ is unbounded and nonconstant, then $\mathcal{N}^{\sigma}_{\infty,k,1}$ is dense in $L^p(\mu)$ for all finite measures μ on $\mathbb{R}^k$.*

Theorem 3.2. *If σ is continuous, bounded and nonconstant, then $\mathcal{N}^{\sigma}_{\infty,k,1}$ is dense in $C([0,1]^d)$.*

The proofs to these theorems can be found in Appendix 7.2.

These results easily generalize to $\mathbb{R}^{d_1}$ valued neural networks, since every component of such a network is itself an $\mathbb{R}$ valued neural network. For more details, see [7].

Next we let $(\mathcal{N}^{\sigma}_{M,d_0,d_1})_M$ be a family of neural networks, such that

$$\begin{aligned}
&1)\ \bigcup_{M\geq 1} \mathcal{N}^{\sigma}_{M,d_0,d_1} = \mathcal{N}^{\sigma}_{\infty,d_0,d_1}.\\
&2)\ \mathcal{N}^{\sigma}_{M,d_0,d_1} \subset \mathcal{N}^{\sigma}_{M+1,d_0,d_1}, \text{for all } M \geq 1\\
&3)\ \text{for all } M \geq 1 \text{ we can write } \mathcal{N}^{\sigma}_{M,d_0,d_1} = \{F^{\theta} \mid \theta \in \Theta_{M,d_0,d_1}\}\\
&\qquad \text{with } \Theta_{M,d_0,d_1} \subset \mathbb{R}^q \text{ for some } q \in \mathbb{N} \text{ depending on } M.
\end{aligned} \tag{3.2}$$

Here θ is a vector of all the possible weights, i.e. the matrix and bias vector entries for the affine functions G_l. Natural choices include $\mathcal{N}^{\sigma}_{M,d_0,d_1}$ being the family of networks with at most M non-zero weights, at most height M or at most depth M. When σ is clear from context we define $\mathcal{N}_{M,d_0,d_1} := \mathcal{N}^{\sigma}_{M,d_0,d_1}$.

3.2 Approximating Hedges

We now define the family of hedges that can be represented by neural networks in a given set $\mathcal{N}_{M,d_0,d_1}$ as

$$\begin{aligned}
\mathcal{H}_M &= \left\{(\delta_k)_{k\leqslant n-1} \in \mathcal{H} \;\middle|\; \delta_k = F_k(I_0,\dots,I_k,\delta_{k-1}), F_k \in \mathcal{N}_{M,r(k+1)+d,d}\right\}\\
&= \left\{(\delta_k)_{k\leqslant n-1} \in \mathcal{H} \;\middle|\; \delta_k = F^{\theta_k}(I_0,\dots,I_k,\delta_{k-1}), \theta_k \in \Theta_{M,r(k+1)+d,d}\right\}
\end{aligned} \tag{3.3}$$

Then we can replace Equation 2.7 with the following variant:

$$\begin{aligned}\pi^M(-Z) &:= \inf_{\delta\in\mathcal{H}_M} \rho(-Z + (\delta\cdot S)_T - C_T(\delta)) \\ &= \inf_{\theta\in\Theta_M} \rho(-Z + (\delta^\theta\cdot S)_T - C_T(\delta^\theta)).\end{aligned} \tag{3.4}$$

Remark 3.3. If S is a Markov process, $Z = g(S_T)$ not path dependent and we work with simple market frictions, then we can write the optimal strategy in the simpler form of $\delta_k = f_k(I_k, \delta_{k-1})$ for some $f_k : \mathbb{R}^{r+d} \to \mathbb{R}^d$. This will be useful for implementing the neural networks later.

Next we show that the strategies in $\mathcal{H}_M$ can approximate the strategies in $\mathcal{H}$, and so by extension the prices $p(Z)$, arbitrarily well. The following result was taken from [3].

Proposition 3.1. *For any $X \in \mathcal{X}$ it holds that*

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

Proof. We first note that the argument δ_{k-1} in F_k in Equation 3.3 is redundant, since δ_{k-1} is itself a function of $I_0, \ldots, I_{k-1}$. For the purpose of this proof we will thus rewrite

$$\mathcal{H}_M = \big\{(\delta_k)_{k\leqslant n-1} \in \mathcal{H} \;\big|\; \delta_k = F_k(I_0,\ldots,I_k), F_k \in \mathcal{N}_{M,r(k+1)+d,d}\big\}.$$

Now since $\mathcal{H}_M \subset \mathcal{H}_{M+1} \subset \mathcal{H}$ for all M, 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 $\pi^M(X) \leqslant \pi(X) + \varepsilon$.

Let $\varepsilon > 0$. By definition, there exists $\delta \in \mathcal{H}$ such that

$$\rho(X + (\delta\cdot S)_T - C_T(\delta)) \leqslant \pi(X) + \frac{\varepsilon}{2}. \tag{3.5}$$

Since δ_k is $\mathcal{F}_k$-measurable there exists some measurable function $f_k : \mathbb{R}^{r(k+1)} \to \mathbb{R}^d$ such that $\delta_k = f_k(I_0, \ldots, I_k)$ for each k. Since Ω is finite δ_k is bounded and so the components of f_k are in $L^1(\mathbb{R}^{r(k+1)}, \mu)$, where μ is the law of $(I_0, \ldots, I_k)$ under $\mathbb{P}$.

Thus we can use Theorem 3.1 to find $F_{k,n} \in \mathcal{N}_{\infty,r(k+1),d}$ such that $F_{k,n}(I)$ converges to $f_k(I)$ in $L^1(\mathbb{P})$ as $n \to \infty$ componentwise. By choosing the right subsequence convergence holds $\mathbb{P}$-a.s. for all k. Writing $\delta_k^n := F_{k,n}(I)$ this implies (since we assumed $\mathbb{P}(\{\omega\}) > 0$ for all ω) that

$$\lim_{n\to\infty} \delta_k^n(\omega) = \delta_k(\omega) \quad \text{for all } \omega \in \Omega. \tag{3.6}$$

Since Ω is finite, ρ can be viewed as a convex function $\rho : \mathbb{R}^N \to \mathbb{R}$. In particular ρ is continuous (see Appendix 7.1). And so

$$\begin{aligned}
&\liminf_{n\to\infty} \rho(X + (\delta^n \cdot S)_T - C_T(\delta^n)) \\
&\quad \leqslant \rho(X + (\delta \cdot S)_T - \limsup_{n\to\infty} C_T(\delta^n)) && \text{continuity of } \rho \\
&\quad \leqslant \rho(X + (\delta \cdot S)_T - C_T(\delta)) && \text{upper semi-continuity of } c_k
\end{aligned}$$

Combining this with (3.5) there exists $n \in \mathbb{N}$ large enough such that

$$\rho(X + (\delta^n \cdot S)_T - C_T(\delta^n)) \leqslant \pi(X) + \varepsilon. \tag{3.7}$$

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

3.3 Numerical Solution for the entropic risk measure

Theorems 3.1, 3.2, and Proposition 3.1 all show that it is possible to build a near-optimal neural network for hedging, but this leaves us with the question of actually finding such a network. To explain the main ideas, we will consider ρ to be the entropic risk measure e_γ, but the same approach works for any sufficiently smooth risk measure.

We start by rewriting (2.7)

$$\pi^M(-Z) = \inf_{\theta\in\Theta_M} \rho\big(- Z + (\delta^\theta \cdot S)_T - C_T(\delta^\theta)\big) = \inf_{\theta\in\Theta_M} J(\theta). \tag{3.8}$$

We assume J to be differentiable, then to find a local minimum we may use the gradient descent algorithm. We start with some initial (random) guess θ_0 and then iterate

$$\theta_{j+1} = \theta_j - \eta_j \nabla J(\theta_j), \tag{3.9}$$

for small $(\eta_j)_{j\in\mathbb{N}}$. Under suitable assumptions θ_j converges to a local minimum of J. This leaves us with two questions, can we find the global minimum, or at the very least a “good” local minimum, and can we calculate ∇J efficiently?

We can answer both questions by using variants of the stochastic gradient descent and back-propagation algorithms. For us this means that we replace the expected value with a sum over a small subset of Ω, called a minibatch, of size $N_b \ll N$ sampled anew for every j. So we replace J in (3.9) with

$$J_j(\theta) = \frac{1}{\gamma} \log \Big(\frac{N}{N_b} \sum_{m=1}^{N_b} \mathbb{P}[\{\omega_m^j\}] \exp \big(-\gamma(-Z(\omega_m^j)+(\delta^\theta\cdot S)_T(\omega_m^j)-C_T(\delta^\theta)(\omega_m^j))\big)\Big), \tag{3.10}$$

where $(\omega_k^j)_{k\leqslant N_{\text{batch}}} \subset \Omega$ are sampled uniformly. This gives the simplest form of the (minibatch) stochastic gradient descent algorithm. It makes computation more efficient simply by considering a smaller sample size. Furthermore this algorithm avoids getting stuck in local minima as each J_j will have slightly different minima, and so θ_j will keep moving.

It is common to not apply log in the objective function J, as it is monotonically increasing and so all the local and global minima stay the same. The notable effect is on the magnitude of the gradient, which is scaled proportionally to the PL_T of the given minibatch, when the log is present. This empirically yielded good results and hence we worked with the objective function as described above. It is worth noting that minibatching will behave differently when the log is still present, since the sum and log do no commute. This is not an issue in this special case as we are trying to find solutions where the objective function is close to zero, which means that the term inside the log will be close to one, and the logarithm is approximately linear close to one, i.e. the log and sum terms will approximately commute.

In order to actually calculate the gradient of J_j we start by naively plugging in. We find

$$\begin{aligned}\nabla_\theta J_j(\theta) &= \frac{1}{\gamma}\frac{\sum_{m=1}^{N_b}\mathbb{P}[\{\omega_m^j\}]\nabla_\theta \exp\big(-\gamma(-Z(\omega_m^j)+(\delta^\theta\cdot S)_T(\omega_m^j)-C_T(\delta^\theta)(\omega_m^j))\big)}{\sum_{m=1}^{N_b}\mathbb{P}[\{\omega_m^j\}]\exp\big(-\gamma\mathrm{PL}_T(\omega_m^j)\big)}\\ &= \frac{1}{\gamma}\frac{\sum_{m=1}^{N_b}\mathbb{P}[\{\omega_m^j\}]\exp\big(-\gamma\mathrm{PL}_T(\omega_m^j)\big)\nabla_\theta\big((\delta^\theta\cdot S)_T(\omega_m^j)-C_T(\delta^\theta)(\omega_m^j)\big)}{\sum_{m=1}^{N_b}\mathbb{P}[\{\omega_m^j\}]\exp\big(-\gamma\mathrm{PL}_T(\omega_m^j)\big)}.\end{aligned} \tag{3.11}$$

We can see that this reduces to calculating the gradient of $(\delta^\theta\cdot S)_T - C_T(\delta^\theta)$, which itself reduces to calculating the gradient of the various feed forward networks F_k. Here we can make use of the inherent structure of neural networks to make this calculation feasible.

Denote by $z^l = A^l a^{l-1} + b^l$ the weighted input of each layer and by a^l the output of the l^{th} layer. Without loss of generality we can assume that the bias terms are zero, since we can append their weights to the matrix and simply append 1 to the input vector. By repeatedly making use of the chain rule we get the following analytic solution:

$$\begin{aligned}\frac{dF_k}{dx}(x) &= \frac{dz^L}{da^{L-1}}(a^{L-1})\frac{da^{L-1}}{dz^{L-1}}(z^{L-1})\cdots\frac{da^1}{dz^1}(z^1)\frac{dz^1}{dx}(x)\\ &= A^L\cdot\sigma'(z^{L-1})\cdot A^{L-1}\cdots\sigma'(z^1)\cdot A^1\end{aligned} \tag{3.12}$$

Backpropagation then consists of evaluating this expression from right to left. To do so we introduce the quantity

$$\begin{aligned} \delta^l &:= \sigma' \cdot (A^{l+1})^T \cdots (A^{L-1})^T \cdot \sigma'(A^L)^T \\ &= \sigma' \cdot (A^{l+1})^T \cdot \delta^{l+1} \end{aligned} \tag{3.13}$$

which can be calculated recursively and efficiently.

For the Markovian and recursive networks (defined in Section 4.1), the output of one time-step becomes the following network's inputs. Hence we have to consider the backpropagation process not just for a single neural network but through multiple networks. This causes no problems and can be done in the same way as described above by seeing the concatenated networks as a single large network which has additional external outputs (i.e. the stock price) every few layers.

Remark 3.4. The specific implementation of stochastic gradient descent we use in all experiments is Adam with the standard parameters, as suggested in the paper, which was introduced in [10].

3.4 The Classical Delta Hedge

Before we consider the hedging results found by the neural network it is worth considering the classical theory. The standard Black-Scholes model is given by the stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{3.14}$$

where W_t is a standard Brownian motion, μ is the annualized drift and σ is the annualized volatility. For some given initial value $S_0 \in \mathbb{R}_{\geq 0}$ we can find the analytic solution

$$S_t = S_0 \exp\left((\mu - \frac{\sigma^2}{2})t + \sigma W_t\right)$$

using basic Itô calculus. Without loss of generality we can set $\mu = 0$ so that S_t is a martingale, since we can use the Girsanov theorem (see Appendix 7.3) to find a martingale measure (also called risk-neutral measure) from the physical measure.

Now let us derive the risk-neutral price and hedge for the binary option. We want to find the value of $\mathbb{E}[f(S_T)]$ where f is defined by (2.1), although the same derivation works for all well behaved f. We now define

$$V_s := \mathbb{E}\big[f(S_T) \mid \mathcal{F}_s\big] = \mathbb{E}\left[f\big(S_t \exp\big(\sigma(W_T - W_t) - \frac{\sigma^2}{2}(T-t)\big)\big) \mid \mathcal{F}_t\right] \tag{3.15}$$

as the value of the option at previous times. Note V_s is a martingale by construction. Furthermore, S_t is $\mathcal{F}_t$-measurable and $W_T - W_t \perp \mathcal{F}_t$. We now write

$$V_s = v(s, S_s) \tag{3.16}$$

for some deterministic function v, which we can do since S_t is a Markov process. We have the following conditions:

$$v(t, 0) = 0 \qquad v(T, x) = f(x) = 1_{x \geq K} \tag{3.17}$$

Now by Itô's formula we find for $v(t, S_t) = V_t$:

$$\begin{aligned} dV_t &= \frac{dv}{dt}(t, S_t)dt + \frac{dv}{dx}(t, S_t)dS_t + \frac{1}{2}\frac{d^2v}{dx^2}(t, S_t)d[S]_t \\ &= \frac{dv}{dt}(t, S_t)dt + \frac{dv}{dx}(t, S_t)dS_t + \frac{1}{2}\frac{d^2v}{dx^2}(t, S_t)\sigma^2 S_t^2 dt \\ &= \frac{dv}{dx}(t, S_t)dS_t + \big(\frac{dv}{dt}(t, S_t) + \frac{1}{2}\frac{d^2v}{dx^2}(t, S_t)\sigma^2 S_t^2\big)dt \\ &= \frac{dv}{dx}(t, S_t)dS_t. \end{aligned} \tag{3.18}$$

Since V_t is a martingale we know that the drift term must be equal to zero. In the last line we instantly see that the hedging strategy is exactly the derivative of the value function with respect to the underlying stock price, also called the Delta.

To now find v we must solve the following PDE with the boundary conditions given by (3.17).

$$\partial_t v(t, x) + \frac{1}{2}\partial_x^2 v(t, x)\sigma^2 x^2 = 0. \tag{3.19}$$

Now let Φ be the cumulative distribution function for the standard normal distribution. Then

$$v(t, x) = \Phi\Big(\frac{\log(\frac{x}{K}) - \frac{\sigma^2}{2}(T - t)}{\sigma\sqrt{T - t}}\Big) =: \Phi(d_-) \tag{3.20}$$

is the solution to (3.19) and so the expected value of the binary option at a given time and stock price. The hedging strategy is then given by the derivative of (3.20) with respect to x. A simple calculation then shows

$$\partial_x v(t, x) = \frac{1}{x\sigma\sqrt{T - t}}\phi(d_-), \tag{3.21}$$

where $\phi = \Phi$‘ is the probability density function of the standard normal distribution.

For the call option the derivation is identical, except for the boundary conditions of the PDE. The solution is then given by

$$\begin{aligned} v(t,x) &= x\Phi(d_+) - K\Phi(d_-), \\ d_+ &:= \frac{\log(\frac{x}{K}) + \frac{\sigma^2}{2}(T-t)}{\sigma\sqrt{T-t}}. \end{aligned} \tag{3.22}$$

The delta for the call option is then simply $\Phi(d_+)$.

3.5 Why Binary Options?

Discontinuities in payoffs lead to sharp deltas around the strike price, especially close to maturity; see Figure 1. This creates no problems in idealized markets where continuous-time trading is possible, and no market frictions exist. However, in real markets, we only have finitely many trading opportunities, market impacts, and liquidity constraints.

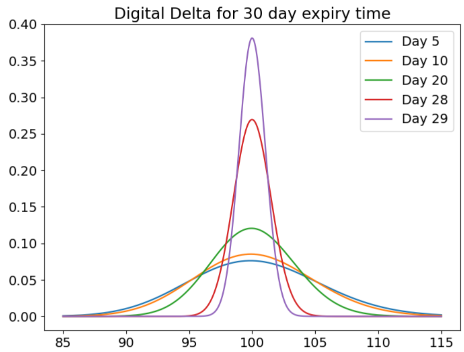


Fig. 1: Current Stock price versus amount of stock owned according s to the delta hedge for various times.

All of these constraints make the rapid trading of large quantities infeasible. Classically, to avoid these problems, binary options were estimated by a call spread with the call strike prices just below and above the binary strike. This will lead to a "smoother" delta, making trading more feasible, but also leads to less efficient hedging, requiring the seller to increase the price. Furthermore, this strategy does not address transaction costs.

Binary options create uncertainty for the seller around the terminal time, which can be observed in Figure 2. The distribution of profits and losses versus

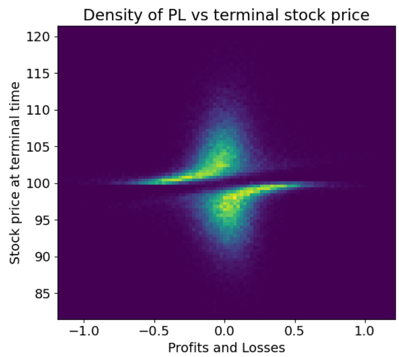


Fig. 2: The joint distribution of the profits & losses and the stock price at terminal time.

the terminal stock price has a significant discontinuity around the strike price, even when hedging. Hence even a slight uncertainty of the terminal stock price leads to a significant uncertainty of profits and losses.

4 Methodology

4.1 Network Structures

Now that we have both a theoretical framework ensuring the existence of good solutions and a numerical tool to find these, we turn to implementation. Like in the structure proposed in [3], we train a separate neural network for every timestep. The neural network for a given time step takes in market information and current holdings and returns the optimal allocation for the next time interval. The three variants we will consider can be seen in Figure 3 below.

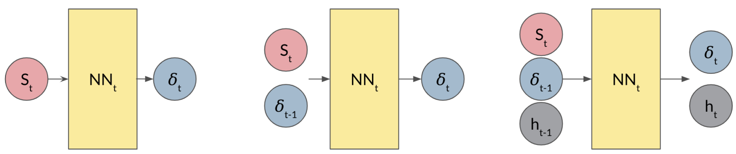


Fig. 3: Three different architectures

We will call a network of the first kind a simple memory-less network, one of the second kind Markovian network, and the third kind a recursive network. The simple network has as an input, for every times step, just the market information for that time step. The Markovian network receives, additionally, its previous output. Finally the recursive network propagates additional information forward in the variable h.

For every kind, the whole network consists of $n-1$ fully connected neural networks, each taking in information as described above. The two hidden layers of these networks use tanh activation (unless otherwise stated), each of which has $8+d$ nodes. This choice is not itself significant as long as the network is large enough to approximate the needed strategies sufficiently.

Intuitively we can think of these networks as being differentiated by how much information can flow from one timestep to the next. For the simple network there is no information flow between time-steps. For the Markovian network only the current holdings are passed on. For the recursive network any information can be passed on given that the hidden dimension is sufficiently large.

We implemented these networks in Tensorflow 2.0. The code is available on the author's GitHub[3].

4.2 Training Details

As has already been stated, since we have transaction costs, the price can not be a trainable variable in the network. So we have to train the network for multiple fixed prices and then choose the price for which $\pi(-Z+p_0)$ is closest to zero, due to Lemma 2.3. The exact details of this approach can be found in Appendix 7.4.

When training, the neural network calculates the profits and losses for every simulated path and then applies the convex risk measure to the minibatch. It then tries to minimize the result using gradient descent as described above. The networks are trained on a deterministic learning rate schedule: every ten epochs the learning rate decreases.

In the special case of no arbitrage and no transaction costs we can, instead of using a risk measure, use mean squared loss and make the price a trainable variable. This will lead the network to find the classical price derived in Section 3.4. This is because we are solving

$$\inf_{\substack{\delta\in\mathcal{H}\\ p\in\mathbb{R}}} \mathbb{E}\big[(f(S_T)-p-(\delta\bullet S)_T)^2\big].$$

If these conditions are not fulfilled a different approach must be taken, since the network can learn to trade superfluously to increase transaction costs and

[3] Link: github.com/nic-kup

simultaneously increase the price to compensate for the extra trading, leading to arbitrary prices.

In the Subsections 5.1 the networks were trained using mean squared error.

5 Results

5.1 No Transaction Costs

First we test the network on binary and call options with no transaction costs. We consider the Black-Scholes market model with no drift, an annualized volatility of

$$\sigma = 0.2$$

and an initial price of $S_0 = 100$. In this market we want to price at-the-money call options, i.e. the strike price is

$$K = 100.$$

From Section 3.4 we know that an at-the-money call option would be worth \$1.868 and a binary option worth \$0.490.

The options we consider mature after twenty days,

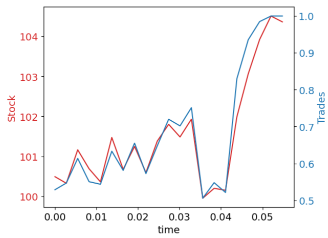
$$T = 20/365,$$

and the network has the ability to trade once a day. This means in particular that our model consists of 20 networks one for every timestep that we trade.

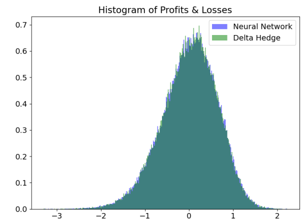
If continuous time trading were possible we could replicate both the European call option and the binary option perfectly. In the current setup of discrete trading this is not possible, however we can still use the Delta hedge described in Section 3.4 for every discrete time step. This still leads to a significant variance reduction compared to not trading, but of course less significant than trading in continuous time.

Simple Models From our derivation of the Delta hedge we would expect the simple network to be able to find an effective strategy in a Black Scholes market without transaction costs. First let us consider a European call option as described in the market above, with a strike price of 100.

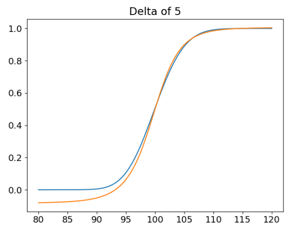
We can see in Figure 4 that the memory-less network learns to approximate the classical Delta hedge very closely. The strategy most accurately approximates the Delta hedge when the underlying is around 100, since most data points will be close to 100 as for any given time the stock price is distributed log-normally with mean 100. The slight deviation in strategy has no noticeable effect on the



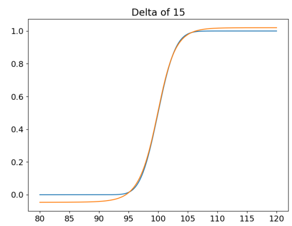
(a) An instance of hedging a call option.



(b) A histogram of outcomes of PL_T.



(c) Delta vs Network strategy.

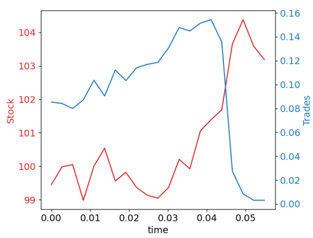


(d) Delta vs Network strategy.

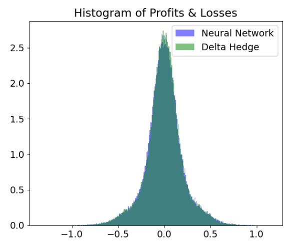
Fig. 4: Results for hedging an ATM call option with a simple network. Found price: $\sim$ \$1.860. For Figures c) and d) the x-axis represents current stock price and y-axis represents the corresponding holdings for that time step. The blue line is the classical hedge, whereas the orange is proposed by the network.

outcome. The histogram of PL_T for the classical Delta hedge and our simple neural network, as overlaid in Figure 4b, are indistinguishable. Furthermore, the price the network found was off by a third of a cent compared to the price of implementing the Delta hedge.

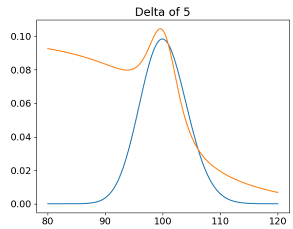
The broad variance in the histogram comes from the size of the time steps. In particular, after the last trading opportunity a whole day passes with no readjustments possibilities.



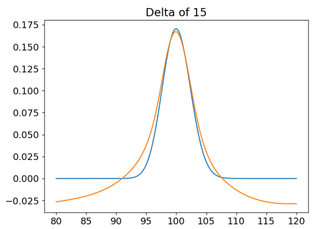
(a) An instance of hedging a binary option.



(b) A histogram of outcomes of PL_T.



(c) Delta vs Network strategy.



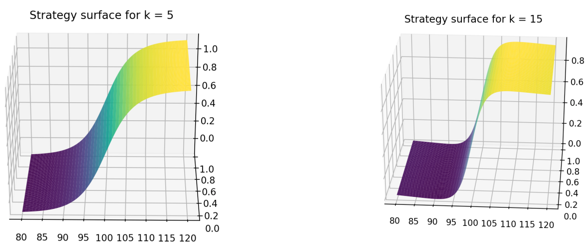
(d) Delta vs Network strategy.

Fig. 5: Results for hedging an ATM binary option with the simple network. Found price: $\sim \$0.488$.

Next let us consider a binary option without transaction costs, again with a strike price of 100. Here we can see in Figure 5 that the approximation of the Delta hedge is significantly worse, especially for the early networks. Despite the large difference in strategy, the profits and losses for the Delta strategy and network strategy are virtually indistinguishable from each other. The price is within one cent of $\mathbb{E}[f(S_T)]$.

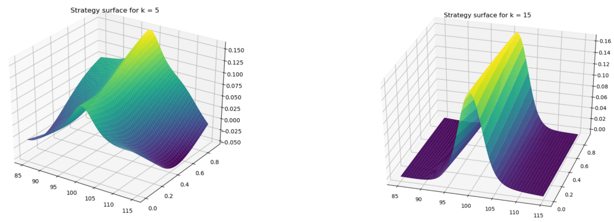
Markovian Models Now we will consider the Markovian models. Since we now have two inputs for each network and a single output we can no longer visualize the strategy with a curve, rather we represent it with a surface. We will train these more general networks on the same options and with the same method as before.

For both the binary and the call option the strategies learned are effectively the same as in the simple case. The networks have mostly learned not to take the previous holdings into account as we would expect. The price and PL_T histograms were effectively identical.



(a) Strategy surface for hedging a call option.

(b) Strategy surface for hedging a call option.



(c) Strategy surface for hedging a binary option.

(d) Strategy surface for hedging a binary option.

Fig. 6: Strategy surface for hedging vanilla call and binary options without transaction costs with a markovian network.

We have managed to recreate the results from [3] and show that the general neural network approach can learn classical solutions to vanilla hedging problems and accurately calculate the corresponding price in complete markets. Any deviations that do appear in the trading strategies seem to have negligible effects on the resulting distribution of PL_T.

5.2 Transaction Costs

We will now consider the same options in a market with transaction costs. No closed form analytic solutions have been found for these types of problems.

There have been algorithmic approaches to this problem such as using tolerance intervals or fixed bandwidths to re-hedge finitely often, see [16].

In this section we consider the entropic risk measure with parameter $\gamma = 1$. This risk measure avoids losses more strongly and so will cause the price to be higher than having the risk measure $\rho(\bullet) = -\mathbb{E}[\bullet]$. We will compare the results to having trained with the entropic risk function without transaction costs. (Call option price without transaction cost is $\sim$ \$2.073 and the same for the binary option is $\sim$ \$0.503.) The hedging strategy the network found for no transaction costs and entropic risk is virtually identical to the strategies found in Section 5.1.

For this Section we will consider proportional transaction costs that are given by the following

$$C_T^{\varepsilon}(\delta) = \sum_{t=1}^{T} \varepsilon S_t |\delta_t - \delta_{t-1}|,$$

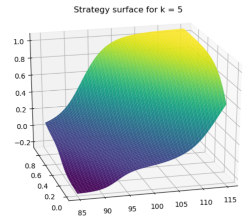
where we set ε to 0.001.

We can see in Figure 7 that the found strategy resembles the standard delta hedge. However for Figure 7a we can see that the strategy is strongly skewed towards the previous holdings. This effect becomes a lot weaker closer to the terminal time, and can not be seen Figure 7b. Compared to trading without transaction costs, the variance of PL_T is also somewhat higher. Notice also in Figure 7d that as the evaluation of the option increases the variance of PL_T also increases. Of course we can also see a negative correlation between option value at final time and PL_T.

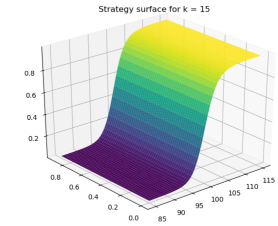
For the binary option with one per mil transaction costs we can see that the trading behaviour, in Figure 8a, is smoothed in the sense that it is less effected by smaller movements, a behaviour that reduces transaction costs. We can also see transaction cost reducing behaviour directly in the strategy surface in Figures 8c and 8d, where holdings in the next trading period are strongly skewed towards the previous holdings, as was the case for the call option.

The profits and losses incurred while trading are considerable more dependent on the evaluation of the option. In Figure 8b we can see the probability densities of the PL_T conditional on the evaluation of the binary option. These two distributions (conditional on $Z = 0$ and $Z = 1$) would ideally be identical, since the seller does not want their profits and losses to be dependent on the option. In the case of no transaction costs the hedge achieved this.

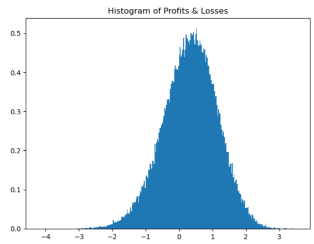
We can see that for both the binary option and the call option the Markovian network learned to implement a skewed version of the Delta Hedge that helps



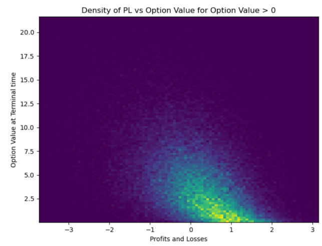
(a) Strategy Surface at time step 5.



(b) Strategy Surface at time step 15.



(c) Histogram of PL_T.



(d) Histogram of PL_T vs Option value.

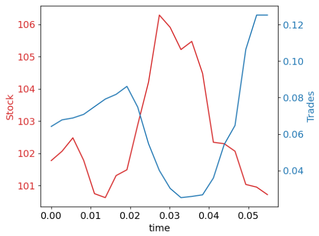
Fig. 7: Results for hedging an ATM European call option with a Markovian network with 1 per mil transaction costs. Found price: $\sim$ \$2.203.

to reduce transaction costs. Next let us compare how the other network models compare to the Markovian network.

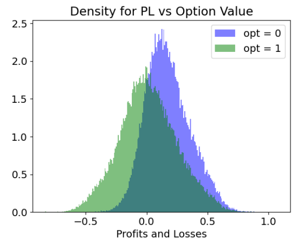
Comparison of Network types Now that we have studied the Markovian network in a little more detail let us see how well it performs in comparison to the other two proposed architectures. We will consider the effect of changing transaction costs and the convex risk measure.

We will first consider a binary option in a Black Scholes market with transaction costs of $\varepsilon = 0.001$, as in Section 5.2. In Figure 9 we can see that the Hidden network performs the best in the sense that the distribution of its PL_T is the sharpest. This means that the replication error is the smallest. We can see in the example that the Markovian and hidden network behave very similarly, whereas the simple network is more sporadic towards the end.

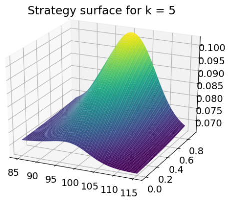
For higher transaction costs we can see in Figure 11b that the PL_T distribution is bi-modal. These two spikes are most distinct for the simple network, since it is the least effective at hedging. We can tell from the example that the simple strategy is also the least dynamic, it relies on a buy and hold strategy.



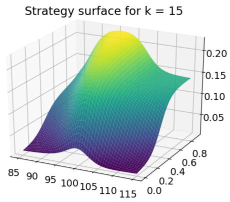
(a) An instance of hedging a binary option.



(b) A histogram of outcomes of PL_T, by option evaluation.

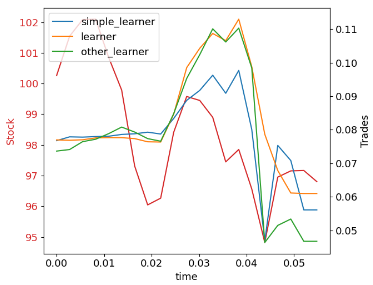


(c) Strategy surface for $k = 5$.

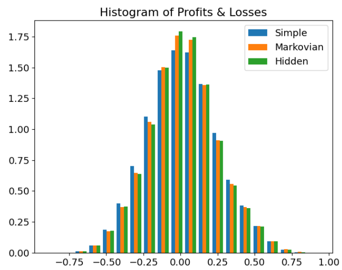


(d) Strategy surface for $k = 15$.

Fig. 8: Results for hedging an ATM binary option with an an auto-regressive network with 1 per mil transaction costs. Found price: $\sim \$0.530$



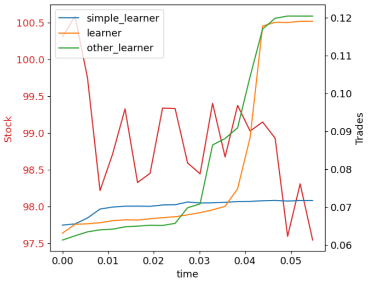
(a) An example of the networks hedging.



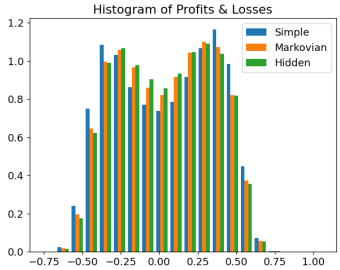
(b) Histogram of PL_T for all network types.

Fig. 9: Hedging a binary option with 1 per mil transaction cost. The simple (simple_learner), Markovian (learner) and recursive (other_learner) networks found the same price of \$0.531.

We also see that increasing the transaction costs naturally increases the price. It is notable that despite the limited capability of the simple network it still managed to find the same price as the other two.

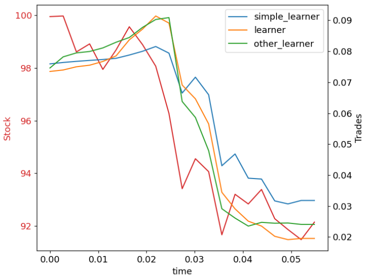


(a) An example of the networks hedging.

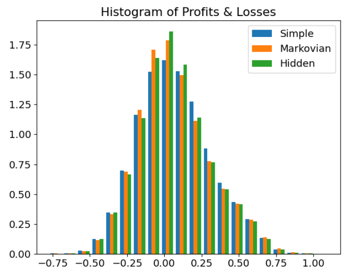


(b) Histogram of PL_T for all network types.

Fig. 10: Hedging a binary option with 3 per mil transaction cost. The simple, Markovian and recursive networks found the same price of $\sim \$0.559$.



(a) An example of the networks hedging.



(b) Histogram of PL_T for all network types.

Fig. 11: Hedging a binary option with 3 per mil transaction cost. The simple, Markovian and recursive networks found \$0.629, \$0.615 and \$0.615 respectively.

Finally let us consider again proportional transaction costs of 3 per mil, i.e. $\varepsilon = 0.003$, but we change the risk measure to the entropic risk measure with $\gamma = 3$. Intuitively a higher gamma value implies a greater aversion of risk, i.e. losses are punished more severely and gains are less valued. We can see that the price increased significantly. In particular the simple network did not manage to find the same price as the Markovian and recursive network, it has to avert risk by raising the price as it is incapable of reducing the variance enough simply by trading. The hidden network still performs the best. Notice that the PL_T distribution has only one peak, but is more strongly skewed to the right.

In Appendix 7.6 we consider hedging the same options as above in the same market, but with the worst case measure.

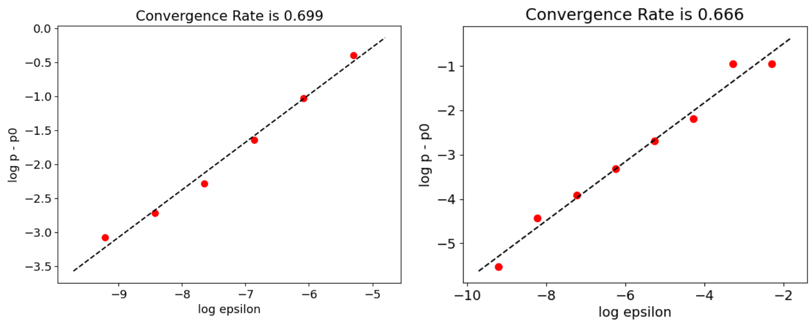
5.3 Convergence of price

Now we are interested in the price as ε goes to zero. The goal is to confirm asymptotics results found by [13] and [9], and to demonstrate the effectiveness of our methodology.

As we can see in these examples the price converges at an exponential rate with respect to the proportional transaction cost. We will consider the Markovian network in the following section as it tends to find the same price as the recursive network, but trains faster. It was shown for call options, and conjectured for all European options, in [13] and [9] respectively, that

$$p_\varepsilon - p_0 = O(\varepsilon^{2/3}), \quad \text{as } \varepsilon \searrow 0. \tag{5.1}$$

Here p_ε is the indifference price of the option with ε as the proportional transaction cost constant. For this example we chose eight equally distanced (in log space) points between $\varepsilon = 0.0001$ and $\varepsilon = 0.1$. For each epsilon we use the Markovian network to calculate the optimal price, and then plot ε versus $p_\varepsilon - p_0$ in a log-log plot.



(a) Call option asymptotics. (b) Binary option asymptotics.

Fig. 12: Price asymptotics of options.

We can see in Figure 12a and 12b that the rate holds remarkably well, despite the numerical sensitivity. Equation 5.1 was shown for continuous time trading where the risk neutral price coincides with the indifference price. Thus we have to compare p_ε to the risk neutral price.

5.4 Heston Model and Trading Restrictions

We will start with a quick reminder of the Heston Model. It is the same as the Black-Scholes model with the only difference being that the volatility is itself driven by a stochastic differential equation:

$$\begin{aligned} dS_t &= \mu S_t dt + \sqrt{V_t} S_T dW_t^0 \\ dV_t &= a(b - \nu_t)dt + \sigma \sqrt{V_t} dW_t^1. \end{aligned} \tag{5.2}$$

Here W^0 and W^1 are two Brownian motions with correlation $\rho \in (-1, 1)$. Usually ρ is negative which means that uncertainty increases in periods where the stock price decreases.

The risk neutral pricing and hedging a European option with payoff $g(S_T)$ at T for some $g : \mathbb{R} \to \mathbb{R}$ in a Heston market can be derived similarly to how we found the closed form solution for the European options in Section 3.4. However, we will not be able to solve it explicitly, i.e. there is no closed form solution. We define $H_t := \mathbb{E}[g(S_T) \mid \mathcal{F}_t]$. By the Markov property we can write $H_t = u(t, S_t, V_t)$, for some

$$u : [0, T] \times [0, \infty)^2 \to \mathbb{R}.$$

We find that

$$g(S_T) = q + (\delta^1 \cdot S)_T + (\delta^2 \cdot V)_T, \tag{5.3}$$

where $q = \mathbb{E}[g(S_T)]$ and

$$\delta_t^1 := \partial_s u(t, S_t, V_t) \text{ and } \delta_t^2 := \frac{\partial_v u(t, S_t, V_y)}{\partial_v L(t, V_t)}. \tag{5.4}$$

Here

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

So in continuous time trading any such option can be replicated perfectly by trading both the stock and the volatility.

Restriction To represent the restriction in trading we replace Equation 3.4 with

$$\pi^M(-Z) = \inf_{\delta \in \mathcal{H}_M^r} \rho(-Z + (\delta \cdot S)_T - C_T(\delta)) \tag{5.5}$$

where

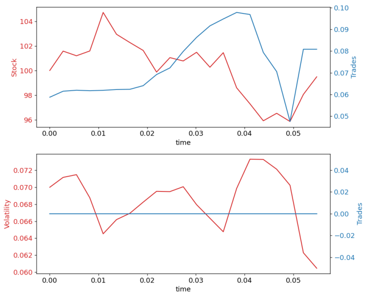
$$\mathcal{H}_M^r = \left\{ H \circ \delta := (H_k(\delta_k))_{k \leqslant n-1} \mid \delta \in \mathcal{H}_M \right\} \tag{5.6}$$

for some restriction functions $H_k : \mathbb{R}^d \to \mathbb{R}^d$. With restriction functions we can represent any restriction given by inequalities. As a simple example we can consider limiting the number of stocks owned to some limit L in a one-dimensional market. Then $H_k(x) = \min(x, L)$ for all k. This can be generalized to restrictions being dependent on past market information, see [3], for our discussion the above framework is sufficient.

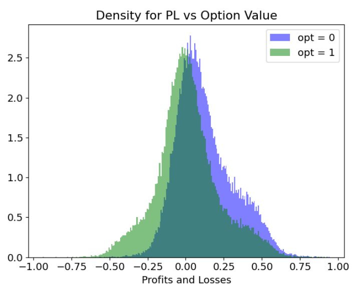
Results In the following examples we consider simply consider the two coordinate projections as possible restriction maps. Specifically this means that we will try to hedge call and binary options by first only trading the underlying, and then by only trading the variance via idealized variance swaps. The parameters chosen for the Heston model are $\mu = 0$, $a = 0.8$, $b = 0.05$, $\rho = -0.65$, $\sigma = 0.2$ $V_0 = 0.07$ and $S_0 = 100$.

Note that we can not fully represent the found strategy in a meaningful way as we have done before since we now have three input dimensions and a single output requiring a four dimensional plot.

Let us first consider hedging a binary option when only trading the underlying. The strategy is very similar to the binary hedge for the Black Scholes model. We can see some similarities to the previously found solutions, including the owned stock increasing for the duration the price stays close to 100.



(a) An example of the networks hedging.

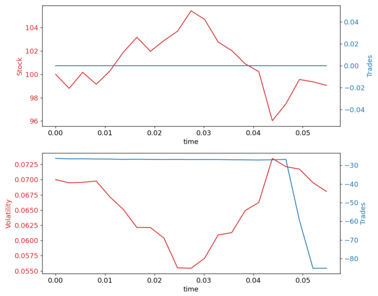


(b) Histogram of PL_T.

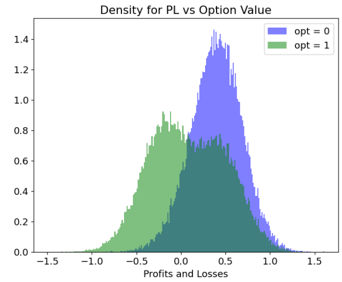
Fig. 13: Hedging a binary option in a Heston market only trading the underlying. Found price is \$0.533

Next let us consider the same option but only trading the variance. As one might expect one can not hedge as effectively in this scenario. This is reflected in the higher price and more significant dissimilarity in the conditional distribution of the PL_T.

We see an interesting phenomenon when only trading the underlying for the call option. We can see in Figure 15b that the shape of the distribution is curved



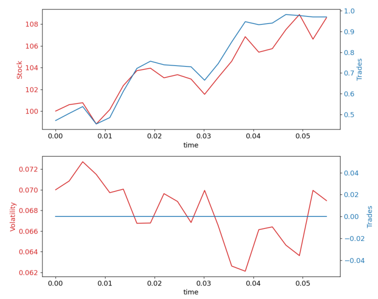
(a) An example of the networks hedging.



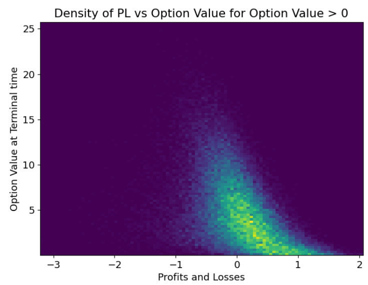
(b) Histogram of PL_T.

Fig. 14: Hedging a binary option in a Heston market only trading the variance. Found price is \$0.563.

as opposed to straight as was the case for the Black Scholes market. This is related to the volatility smile.



(a) An example of the networks hedging.



(b) Histogram of PL_T

Fig. 15: Hedging a call option in a Heston Market only trading the underlying. Found price is \$2.660.

6 Conclusion

The capacity of deep approaches to learn effective hedges, and thus reasonable prices, for various options, was tested under convex risk measures. We found that in situations where analytic solutions are known, the deep approach manages to approximate these solutions well, and in more complicated situations, the network still finds reasonable solutions. In particular, for transaction costs, we found that the price of hedging both European call and binary options converges to the price without transaction costs exponentially fast at the conjectured rate. The discontinuity of the binary options seems to have no impact on the performance of the networks.

We saw that network architecture has a significant impact on the effectiveness of the found strategy. The recursive network led to better results than the Markovian, which led to better results than the simple network. This effect can be observed for both the call and the binary option. The improvement was most significant between the simple and Markovian networks. This was seen in the distribution of profits and losses, the price, and also visually in the examples.

Next, we tested the convergence of the price as the proportional transaction costs tend to zero. Our results support the conjecture that the convergence rate is $2/3$ for all European options for an unrestricted Black Scholes market. This result also held for binary options.

Finally, we looked at incomplete markets and restricted trading in a Heston market. We found for restricting to the underlying the network found a solution similar to the standard Black-Scholes delta hedge, as one might expect. More surprisingly, the network also found an effective strategy for hedging a binary option when only trading the volatility.

A topic that could be explored in more detail include the effect on different discretization of time. In particular letting the network search for an optimal discretization. Further we can look at convergence properties in more detail, both from the theoretical and numerical / experimental point of view. The factor of $2/3$ for the speed of convergence has only been proved for a narrow case of continuous time trading. We could investigate the effects of network architecture, discretization and market model on the rate.

More generally it is worth exploring path dependent options such as American options in more detail. The stopping problem for such options can of course also be solved with deep approaches, see e.g. [2]. Lastly, we can consider not just trading stocks, but also incorporating buying vanilla options into our deep framework, i.e. taking a deep approach to [8].

References

1. ARROW, K.J.: The theory of risk-bearing: Small and great risks. Journal of Risk and Uncertainty **12**(2/3), 103–111 (1996), http://www.jstor.org/stable/41760802
2. Becker, S., Cheridito, P., Jentzen, A.: Deep optimal stopping (2018)
3. Bühler, H., Gonon, L., Teichmann, J., Wood, B.: Deep Hedging. arXiv e-prints arXiv:1802.03042 (Feb 2018)
4. Davis, M., Panas, V., Zariphopoulou, T.: European option pricing with transaction costs. Siam Journal on Control and Optimization - SIAM J CONTR OPTIMIZAT **31** (03 1993). https://doi.org/10.1137/0331022
5. Föllmer, H., Schied, A.: Stochastic Finance: An Introduction in Discrete Time. De Gruyter Textbook, De Gruyter (2016)
6. Friedman, A.: Foundations of Modern Analysis. Dover Books on Mathematics Series, Dover (1982), https://books.google.ch/books?id=yT56SqF0xpoC
7. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Netw. **4**(2), 251–257 (Mar 1991). https://doi.org/10.1016/0893-6080(91)90009-T, https://doi.org/10.1016/0893-6080(91)90009-T
8. Ilhan, A., Jonsson, M., Sircar, R.: Optimal static-dynamic hedges for exotic options under convex risk measures. Stochastic Processes and their Applications **119**(10), 3608–3632 (October 2009)
9. Kallsen, J., Muhle-Karbe, J.: Option pricing and hedging with small transaction costs. Mathematical Finance **25**(4), 702–723 (2015). https://doi.org/10.1111/mafi.12035, https://onlinelibrary.wiley.com/doi/abs/10.1111/mafi.12035
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014)
11. Leland, H.E.: Option pricing and replication with transactions costs. The Journal of Finance **40**(5), 1283–1301 (1985), http://www.jstor.org/stable/2328113
12. Pratt, J.W.: Risk aversion in the small and in the large. Econometrica **32**(1/2), 122–136 (1964), http://www.jstor.org/stable/1913738
13. Rogers, L.: Why is the effect of proportional transaction costs is O($\delta^{2/3}$), pp. 303–308 (01 2004)
14. Soner, H.M., Barles, G.: Option pricing with transaction costs and a nonlinear Black-Scholes equation. Finance and Stochastics **2**(4), 369–397 (1998), https://link.springer.com/article/10.1007/s007800050046
15. Werner, W.: Lecture notes in brownian motion and stochastic calculus (April 2018)
16. Zakamulin, V.: The best hedging strategy in the presence of transaction costs. International Journal of Theoretical and Applied Finance (IJTAF) **12**, 833–860 (09 2009). https://doi.org/10.1142/S0219024909005488

7 Appendix

7.1 Continuity of ρ

We can view $\rho : \mathcal{X} \to \mathbb{R}$ as a function from $\mathbb{R}^N \to \mathbb{R}$ instead by identifying $\mathcal{X} \cong \mathbb{R}^N$. This can easily be done by identifying every element of Ω with a dimension of the space $\mathbb{R}^N$.

Lemma 7.1. A convex risk measure ρ on a finite probability space Ω with N elements is continuous, when seen as a function from $\mathbb{R}^N \to \mathbb{R}$.

Proof. Let $X \in \mathbb{R}^N$ and $\varepsilon > 0$. Now let $Y \in B_{\varepsilon/2}(0)$. Let $Y_{\max} = \max_i |Y^i|$. Clearly the $Y_{\max}$ is bounded by $\varepsilon/2$. By cash-invariance and monotonicity we find that

$$\rho(X) - Y_{\max} = \rho(X + Y_{\max} \cdot 1) \leqslant \rho(X + Y) \leqslant \rho(X - Y_{\max}) = \rho(X) + Y_{\max}. \tag{7.1}$$

And so it follows that

$$\big|\rho(X) - \rho(X + Y)\big| \leqslant 2Y_{\max} \leqslant 2\varepsilon/2 = \varepsilon. \tag{7.2}$$

■

7.2 Proof of Universal Approximation Theorem

The following proof is taken from [7].

Proof (of Theorems 3.1 and 3.2). We will show the universal approximation property for neural networks with a single hidden layer. So let us define the networks with n hidden units as

$$\mathfrak{N}_k^n(\sigma) := \big\{h : \mathbb{R}^k \to \mathbb{R} \mid h(x) = \sum_{j \leqslant n} \beta_j \sigma(\langle a_j, x\rangle - \theta_j)\big\}. \tag{7.3}$$

We can then define all possible single hidden layer networks as

$$\mathfrak{N}_k(\sigma) := \bigcup_{n=1}^{\infty} \mathfrak{N}_k^n(\sigma) \tag{7.4}$$

As σ is bounded, $\mathfrak{N}_k(\sigma)$ is a linear subspace of $L^p(\mu)$ for all finite measures μ on $\mathbb{R}^k$. If, for some μ, $\mathfrak{N}_k(\sigma)$ is not dense in $L^p(\mu)$, there is, by Lemma 7.2, some nonzero continuous linear functional Λ on $L^p(\mu)$ that vanishes on $\mathfrak{N}_k(\sigma)$.

We can write this as $\Lambda(f) = \int_{\mathbb{R}^k} fg d\mu$ for some $g \in L^q(\mu)$, where q is the conjugate of p. Define $\psi(B) := \int_B g d\mu$, then by Hölder's inequality we find

$$|\psi(B)| \leqslant \|1_B\|_{p,\mu} \|g\|_{q,\mu} \leqslant \mu(\mathbb{R}^k)^{\frac{1}{p}} \|g\|_{q,\mu} < \infty, \tag{7.5}$$

and so ψ is a nonzero finite signed measure on $\mathbb{R}^k$. In particular $\Lambda(f) = \int_{\mathbb{R}^k} fgd\mu = \int_{\mathbb{R}^k} fd\psi$. Since Λ vanishes on $\mathfrak{N}_k(\sigma)$, we can conclude that

$$\int_{\mathbb{R}^k} \sigma(\langle a, x\rangle - \theta) d\sigma(x) = 0, \tag{7.6}$$

for all $a \in \mathbb{R}^k$ and $\theta \in \mathbb{R}$. Now the question becomes can there exist such a non-zero signed measure. We call an activation function *discriminatory* if no nonzero finite signed measure ψ exists such that (7.6) holds for all a and θ.

So now we want to show that if σ is bounded and nonconstant, it is discriminatory. So assume ψ is a nonzero signed measure such that (7.6) holds. Fix some $u \in \mathbb{R}^k$ and let ψ_u be the finite signed measure on $\mathbb{R}$ induced by $x \mapsto \langle u, x\rangle$, that is

$$\psi_u(B) = \psi(\{x \in \mathbb{R}^k : \langle u, x\rangle \in B\}).$$

Then for all bounded function ξ on $\mathbb{R}$ we have

$$\int_{\mathbb{R}^k} \xi(\langle u, x\rangle) d\psi = \int_{\mathbb{R}} \xi(t) d\psi_u(t).$$

Hence by assumption

$$\int_{\mathbb{R}^k} \sigma(\lambda\langle u, x\rangle - \theta) d\psi(x) = \int_{\mathbb{R}} \sigma(\lambda t - \theta) d\psi_u(t) = 0$$

for all $\lambda, \theta \in \mathbb{R}$.

To simplify notation, we will denote by $L := L^1(\mathbb{R})$ the Lebesgue integrable functions and by $M = M(\mathbb{R})$ the space of finite signed measures on $\mathbb{R}$. For $f \in L$, $\|f\|_L$ denotes the usual L^1 norm and $\hat{f}$ the Fourier transform. Similarly, for $\tau \in M$, $\|\tau\|_M$ denotes the total variation of τ on $\mathbb{R}$ and $\hat{\tau}$ the Fourier transform.

Choosing θ such that $\sigma(-\theta) \neq 0$ and setting $\lambda = 0$, we find that in particular $\int_{\mathbb{R}} d\psi(t) = \hat{\psi}_u(0) = 0$. For $u = 0$, ψ_u is concentrated at $t = 0$. But $\psi_0(\{0\}) = \hat{\psi}_0 = 0$. Hence $\psi_0 = 0$. Now suppose $u \neq 0$. Pick any function $f \in L$ whose Fourier transform has no zero (e.g. $f(t) = \exp(-t^2)$). Now consider

$$\int_{\mathbb{R}} \int_{\mathbb{R}} \sigma(\lambda(s + t) - \theta) f(s)\ ds\ d\psi_u(t). \tag{7.7}$$

As

$$\int_{\mathbb{R}} \int_{\mathbb{R}} |\sigma(\lambda(s + t) - \theta)||f(s)|\ ds\ d|\psi_u|(t) \leqslant \|f\|_L \|\psi_u\|_M \sup_{r \in \mathbb{R}} |\sigma(r)| < \infty,$$

we can use Fubini.

$$
\begin{aligned}
0 &= \int_{\mathbb{R}} \left[\int_{\mathbb{R}} \sigma(\lambda t - (\theta - \lambda s)) d\psi_u(t) \right] f(s) ds \\
&= \int_{\mathbb{R}} \int_{\mathbb{R}} \sigma(\lambda(s+t) - \theta) f(s) \; ds \; d\psi_u(t) \\
&= \int_{\mathbb{R}} \sigma(\lambda t - \theta) \; d(f * \psi_u)(t),
\end{aligned} \tag{7.8}
$$

where $f * \psi_u$ denotes the convolution of f and ψ_u. Since $f * \psi_u$ is absolutely continuous with respect to the Lebesgue measure we can denote by $h \in L$ the Radon-Nikodym derivative. Then $\hat{h} = \hat{f}\hat{\psi}_u$, hence $\hat{h}(0) = 0$.

This is equivalent to $\int_{\mathbb{R}} \sigma(\lambda t - \theta) h(t) dt = 0$. Let $\alpha \neq 0$ and $\gamma \in \mathbb{R}$. By setting $\lambda = \frac{1}{\alpha}$ and $\theta = -\frac{\gamma}{\alpha}$ and then substituting $t \mapsto \alpha t - \gamma$, we find that for all γ and $\alpha \neq 0$ that

$$
\int_{\mathbb{R}} \sigma(t) h(\alpha t - \gamma) dt = 0
$$

Let us write $M_\alpha h(t)$ for $h(\alpha t)$. The above equation implies $\int_{\mathbb{R}} \sigma(t) g(t) dt$ vanishes for all $g \in I$ where $I \subset L$ is the subspace spanned by the family $M_\alpha h$ with $\alpha \neq 0$. I is an ideal in L.

Let us denote by $Z(g)$ the set of all $\omega \in \mathbb{R}$ where the Fourier transform $\hat{g}(\omega)$ for some $g \in L$ vanishes. If I is an ideal of L define by $Z(I)$, the zero set of I, as the set of ω where the Fourier transforms of all functions in I vanish.

Suppose h is nonzero, as $\widehat{M_\alpha h}(\omega) = \frac{1}{\alpha}\hat{h}(\frac{\omega}{\alpha})$ we find $Z(I) = \{0\}$. In face I is precisely the set of all integrable functions g with $\int_{\mathbb{R}} g(t) dt = \hat{g}(0) = 0$. To see this let us first note that for all function $g \in I$ we trivially have $\{0\} = Z(I) \subset Z(f)$. Conversely, suppose that g has zero integral. As the intersection of the boundaries of $Z(I)$ and $Z(g)$ equals $\{0\}$ and hence contains no perfect set, implies that $g \in I$.

Hence the integral $\int_{\mathbb{R}} \sigma(t) g(t) dt$ vanishes for all integrable functions which have zero integral. It is easily seen that implies that σ is constant, which is false by assumption. Thus $h \equiv 0$ and also $\hat{h} = \hat{f}\hat{\psi}_u \equiv 0$, which in turn implies $\hat{\psi}_u \equiv 0$, since $\hat{f}$ has no zeros by assumption. Hence $\psi_u \equiv 0$.

Since ψ_u is identically zero for all $u \in \mathbb{R}^k$ we complete the proof by explicitly calculating the Fourier transform of ψ at u. Then

$$
\begin{aligned}
\hat{\psi}(u) &= \int_{\mathbb{R}^k} \exp(i\langle u, x\rangle) d\psi(x) \\
&= \int_{\mathbb{R}} \exp(it) d\psi_u(t) \\
&= 0.
\end{aligned} \tag{7.9}
$$

And so $\psi \equiv 0$. Hence σ must be discriminatory.

■

The following Lemma with proof was taken from [6].

Lemma 7.2. *Let Y be a linear subspace of a normed linear space X. If Y is not dense in X, then there exists a functional $f \neq 0$ such that $f(y) = 0$ for all $y \in Y$.*

Proof. Indeed if $\bar{Y} \neq X$, then there is a point x_0 satisfying

$$\inf_{y \in Y} \|y - x_0\| = d > 0. \tag{7.10}$$

Now denote by Y_1 the linear space spanned by Y and x_0. Since $x_0 \notin Y$, every point x in Y_1 has the form $x = y + \lambda x_0$, where $y \in Y$ and the scalar λ are uniquely determined. Consider the linear functional g defined by $g(y + \lambda x_0) = \lambda$. In particular g is zero on Y. If $\lambda \neq 0$, then

$$\|y + \lambda x_0\| = |\lambda| \Big\| \frac{y}{\lambda} + x_0 \Big\| \geq |\lambda| d. \tag{7.11}$$

Hence $|g(x)| \leqslant \frac{\|x\|}{d}$ for all $x \in Y_1$. In particular $\|g\| \leqslant \frac{1}{d}$. Now let $(y_n)_n \subset Y$ be a sequence s.t. $\|x_0 - y_n\| \to d$. Then

$$1 = g(x_0 - y_n) \leqslant \|g\| \|x_0 - y_n\| \to \|g\| d. \tag{7.12}$$

And so $\|g\| = \frac{1}{d}$. Then by the Hahn-Banach Theorem we can extend g to all of X. ■

7.3 Girsanov Theorem

Theorem 7.3 (Girsanov). We consider a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_t, P)$. Let N denote a continuous martingale started from $N_0 = 0$. Now assume that exponential martingale $\mathcal{E}_t$ is uniformly integrable and we denote its limit by $\mathcal{E}_\infty$. Define a new probability measure Q on $\mathcal{F}$ by

$$Q(A) = \mathbb{E}[1_A \mathcal{E}_\infty].$$

Then for any continuous local martingale $M = (M_t)_t$ the process

$$\tilde{M_t} := M_t - [M, N]_t$$

is a local martingale under the probability measure Q.

The following proof was taken from my notes taken in [15].

Proof. It suffices to prove the statement when $M_0 = 0$ almost surely (else we replace $(M_t)_t$ by $(M_t - M_0)_t$). We will work under the probability measure P. The processes $\mathcal{E} = \mathcal{E}(N)$ and $\tilde{M}$ are semimartingales, so we can apply Itô's formula to the semimartingale $U_t = (\mathcal{E}_t \tilde{M}_t)_t$.

Recall that $d\mathcal{E}_t = \mathcal{E}_t dN_t$ and $d[\tilde{M}, N]_t = d[M, N]_t$. Then

$$\begin{aligned} dU_t &= \tilde{M}_t d\mathcal{E}_t + \mathcal{E}(dM_t - d[M, N]_t) + d[\mathcal{E}, M]_t \\ &= \tilde{M}_t d\mathcal{E}_t + \mathcal{E}_t dM_t. \end{aligned} \tag{7.13}$$

Hence U is a local martingale. Next we define the stopping time

$$T_n := \inf(t \geqslant 0 \mid \tilde{M}_t \notin [-n, n] \text{ or } U_t \notin [-n, n]).$$

We see that $T_n \to \infty$ almost surely since U and $\tilde{M}$ are continuous. Our goal is to show that $\tilde{M}^{T_n}$ is an $(\mathcal{F}_t)$-martingale under Q, which implies that $\tilde{M}$ is a local martingale under Q. It holds that $|\tilde{M}^{T_n}| \leqslant n$, and so integrable with respect to Q. This means we have to check that for all $s, t \geqslant 0$ and $A \in \mathcal{F}_t$ that

$$\mathbb{E}_Q[\tilde{M}_t^{T_n} 1_A] = \mathbb{E}_Q[\tilde{M}_{t+s}^{T_n} 1_A].$$

In the case when $T_n \leqslant t$ we find that $\tilde{M}_t^{T_n} = \tilde{M}_{t+s}^{T_n} = \tilde{M}_{T_n}$, which implies

$$\mathbb{E}_Q[\tilde{M}_t^{T_n} 1_A 1_{T_n \leqslant t}] = \mathbb{E}_Q[\tilde{M}_{T_n} 1_A 1_{T_n \leqslant t}] = \mathbb{E}_Q[\tilde{M}_{t+s}^{T_n} 1_A 1_{T_n \leqslant t}].$$

By definition of T_n we have that U^{T_n} is a bounded martingale for P and so

$$\mathbb{E}_P[\tilde{M}_t^{T_n} \mathcal{E}_t^{T_n} 1_{A \cap \{t < T_n\}}] = \mathbb{E}_P[\tilde{M}_{t+s}^{T_n} \mathcal{E}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}]. \tag{7.14}$$

Now consider that $\tilde{M}_t^{T_n} 1_{A \cap \{t < T_n\}}$ is a bounded $\mathcal{F}_t$-measurable random variable, which, using $\mathcal{E}_t = \mathbb{E}_P[\mathcal{E}_\infty | \mathcal{F}_\infty]$, implies

$$\begin{aligned} \mathbb{E}_P[\tilde{M}_t^{T_n} \mathcal{E}_t^{T_n} 1_{A \cap \{t < T_n\}}] &= \mathbb{E}_P[\mathcal{E}_t \tilde{M}_t 1_{A \cap \{t < T_n\}}] \\ = \mathbb{E}_P[\mathcal{E}_\infty \tilde{M}_t 1_{A \cap \{t < T_n\}}] &= \mathbb{E}_Q[\tilde{M}_t 1_{A \cap \{t < T_n\}}]. \end{aligned} \tag{7.15}$$

If, on the other hand, we define $S = \min(t + s, T_n)$ we can easily check that $A \cap \{t < T_n\} = A \cap \{t < S\} \in \mathcal{F}_S$, so that $\tilde{M}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}$ is an $\mathcal{F}_S$-measurable random variable. Hence

$$\begin{aligned} \mathbb{E}_P[\mathcal{E}_{t+s}^{T_n} \tilde{M}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}] &= \mathbb{E}_P[\mathcal{E}_S \tilde{M}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}] \\ = \mathbb{E}_P[\mathcal{E}_\infty \tilde{M}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}] &= \mathbb{E}_Q[\tilde{M}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}] \end{aligned} \tag{7.16}$$

Combining equations (7.14), (7.15) and (7.16) we find that:

$$\begin{aligned} \mathbb{E}_Q[\tilde{M}_t 1_{A \cap \{t < T_n\}}] &= \mathbb{E}_P[\tilde{M}_t^{T_n} \mathcal{E}_t^{T_n} 1_{A \cap \{t < T_n\}}] \\ = \mathbb{E}_P[\tilde{M}_{t+s}^{T_n} \mathcal{E}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}] &= \mathbb{E}_Q[\tilde{M}_{t+s}^{T_n} 1_{A \cap \{t < T_n\}}]. \end{aligned}$$

■

Now we can apply this to the Black-Scholes model. Suppose we have a stock price S_t given by

$$dS_t = S_t(\mu dt + \sigma dW_t),$$

for $\mu \in \mathbb{R}$, $\sigma \in \mathbb{R}_{>0}$ and W a Wiener process. We know that we can write

$$S_t = S_0 \exp\big((\mu - \frac{\sigma^2}{2})t + \sigma W_t)\big) = \mathcal{E}(\sigma W_\bullet + \mu \bullet)_t.$$

So if we can find a measure Q such that $\sigma W_t + \mu t$ is a martingale on $[0, T]$ then S_t will also be martingale on the same interval.

We can suppose that filtration $\mathcal{F}_t$ is generated by a Brownian motion. Then, by a corollary of the Girsanov theorem, candidates for Q can be represented by

$$Z_t = \frac{dQ}{dP}\Big|_t = \exp\big(\int_0^t \lambda_s dW_s - \frac{1}{2}\int_0^t \lambda_s^2 ds\big),$$

for some continuous process λ. Note that Z_t is a true P-martingale on $[0, T]$. Furthermore, $\tilde{W}_t = W_t - \int_0^t \lambda_s ds$ is a Q-Brownian motion. By setting $\lambda_t \cong \frac{\mu}{\sigma}$ we find that

$$S_t = \mathcal{E}(\sigma W_\bullet + \mu \bullet)_t = \mathcal{E}(\sigma \tilde{W}_{\bullet t})$$

which is exactly a Q martingale.

7.4 Searching through Prices

To optimize the searching through possible prices, we implemented two ideas. First, going through prices in a tree-like manner, and secondly retraining the same model instead of initializing a new model for every price.

First, let us consider the tree search. To show why this works we need that fact that

$$\hat{\pi}_p(-Z) := \inf_{\delta \in \mathcal{H}} E[\rho(-Z + (\delta \cdot S)_T + p - C_T(\delta))]$$

is monotonically increasing as a function of the price p. This allows us to skip over prices by only ever considering the largest price that is still negative. To be exact, consider we want to search through prices between p_0 and $p_0 + I_0$ for $p_0 \in \mathbb{R}$ and $I_0 \in \mathbb{R}_{>0}$. Then for some $n \in \mathbb{N}$ we consider the prices $\{p_0, p_0 + \frac{1}{n}I, \ldots, p_0 + \frac{n-1}{n}I\}$, and for each we calculate the loss. We then consider the first price with non-negative loss and call it p_1. We now set $I_1 = \frac{I_0}{n}$, and then repeat this process until we have calculated the price to our desired accuracy. The code above uses $n = 4$.

Second, we do not initialize a new network for every price, but simply retrain the weights that we have learned from the previous price. This assumes that there is a near optimal neural network $F^{\theta_{p+\varepsilon}}$ close to F^{θ_p} for a price $p + \varepsilon$ close to p.

Note that this does not require the map $p \mapsto \arg\min_\delta E[\rho(-Z + (\delta \bullet S)_T + p - C_T(\delta))]$ to be continuous.

This approach leads to the algorithm having three distinct training loops. The first loop trains with price p_0 to then make it easier to train for each of the prices. The second loop iterates through the prices in the way described above and then retrains the model for each. Here we train less for each price than the first loop because of our assumption that a good solution is near (in the space of neural network weights). Finally, we retrain on the final price to get the right strategy for it.

7.5 Notes on American Options

American call options are a prime example of path dependent options. They work just as European options do, but instead the buyer has the right to exercise the option at any time before the expiration date. So the payoff is given by

$$f((S_t)_t) = \max(0, S_\tau - K),$$

where τ is some stopping time representing the buyers choice to exercise the option.

Before we can price American options we must first be able to evaluate the above expression. To do so we will assume that the buyer will try to exercise the option optimally. This means we want to optimize

$$\max_{\tau \in \mathcal{T}} \mathbb{E}\big[\max(0, S_\tau - K)\big],$$

where $\mathcal{T}$ is the set of all stopping times. To do so we will use the methods from [2]. The authors describe a method to train a deep neural network to be a near optimal stopping time that we will call τ^*. The deep hedging framework accepts $f^*((S_t)_t) = \max(0, S_{\tau^*} - K)$ as a valid claim to hedge against.

Now let us consider the results in a Black-Scholes market with an annualized volatility of $\sigma = 0.2$, a strike price of $K = 100$ and a drift of $\mu = -0.6$. The expiration time is $T = 20/365$. We also have transaction costs of $\varepsilon = 0.001$. We will train the recursive network to learn the hedge. First let us look at the stopping algorithm.

We can see that the distribution of stops has significant jumps, that are due to the nature of the stopping time. These jumps are however not see in the density of PL_T in Figure 17a. We can tell from the examples that the network has an understanding of when the buyer exercises the option (even though it is *not* explicitly given this information) as it halts trading activity after this event.

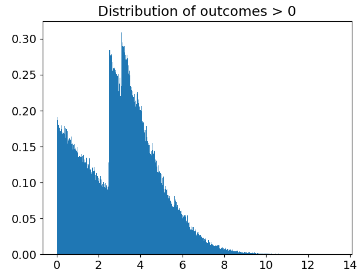
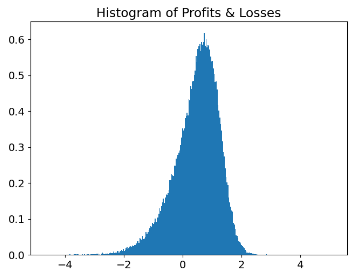
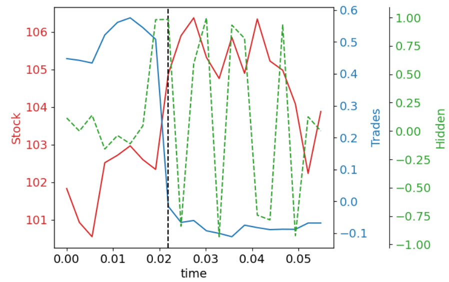


Fig. 16: The density of $\max(0, S_{\tau^*} - K) \mid \max(0, S_{\tau^*} - K) > 0$.



(a) The density of PL_T.



(b) An example hedge: The dotted black line indicates where the buyer exercised, the dotted green line indicates the hidden state.

Fig. 17: Hedging an American option. Found price: \$2.0936.

We can even see in Figure 18 that trading continues until the expiry date, since the option is never exercised.

7.6 Worst Case Measure

An interesting phenomena appears when training with the worst case measure. The interpretation of training with this risk measure is avoiding a loss for the worst case scenario. We see the effect of this in the histogram of Figure 19, specifically none of the the examples are below zero, i.e. for every outcome the seller of the option turns a profit. The strategy for hedging is still very familiar.

For the binary option we still see that there are no outcomes ending in loss for the seller. However the shape of both the histogram and the strategy are radically changed.

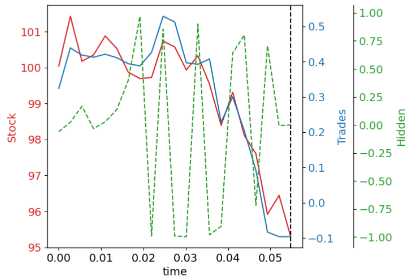
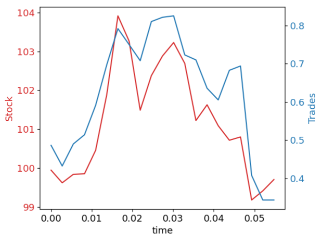
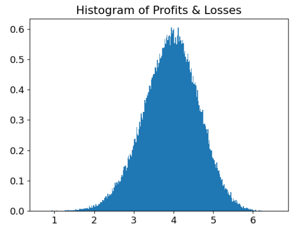


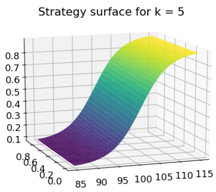
Fig. 18: Another example.



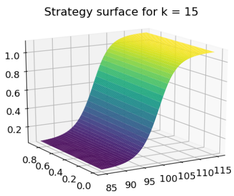
(a) An instance of hedging a binary option.



(b) A histogram of outcomes of PL_T, by option evaluation.



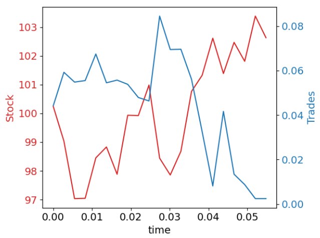
(c) Strategy surface for $k = 5$.



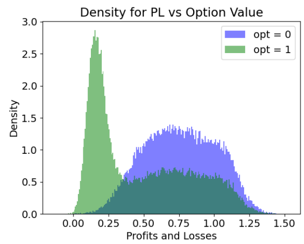
(d) Strategy surface for $k = 15$.

Fig. 19: Results for hedging an ATM binary option with an an auto-regressive network with 1 per mil transaction costs. Found price: $\sim \$5.80$

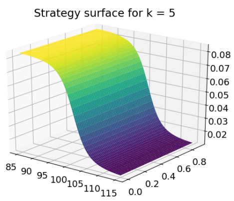
It is notable that in training every update step only ever saw a single training example, i.e. the stock movement that lead to the largest loss, simply by the nature of the choice of convex risk measure, yet the network still managed to learn effective strategies.



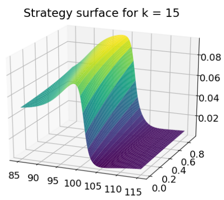
(a) An instance of hedging a binary option.



(b) A histogram of outcomes of PL_T, by option evaluation.



(c) Strategy surface for $k = 5$.



(d) Strategy surface for $k = 15$.

Fig. 20: Results for hedging an ATM binary option with an an auto-regressive network with 1 per mil transaction costs. Found price: $\sim \$0.994$