Machine Learning
- Option Pricing, Calibration, Hedging -

- Deep Hedging -

WBS Quant Conference - Workshop, Rome 2019

Jörg Kienitz / Nikolai Nowaczyk, Quaternion

UCT, BUW, Finciraptor finciraptor.de, joerg.kienitz@gmx.de

Disclaimer

This presentation and any accompanying material are being provided solely for information and general illustrative purposes. The author will not be responsible for the consequences of reliance upon any information contained in or derived from the presentation or for any omission of information therefrom and hereby excludes all liability for loss or damage (including, without limitation, direct, indirect, foreseeable, or consequential loss or damage and including loss or profit and even if advised of the possibility of such damages or if such damages were foreseeable) that may be incurred or suffered by any person in connection with the presentation, including (without limitation) for the consequences of reliance upon any results derived therefrom or any error or omission whether negligent or not. No representation or warranty is made or given by the author that the presentation or any content thereof will be error free, updated, complete or that inaccuracies, errors or defects will be corrected.

The views are solely that of the author and not of his employer, Quaternion Risk Management GmbH. The Chatham House rules apply.

The presentation may not be reproduced in whole or part or delivered to any other person without prior permission of the author.

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Hedging & Deep Learning

Hedging as Sell Side Business Model

- Issue a derivative X and sell to clients
- Replicate payoff of derivative X by a portfolio with a trading strategy ϕ that only trades in the underlying S
- Being long the portfolio and short the derivative is (theoretically) riskless $\Longrightarrow$ make profits in fees

Deep Learning

- Machine Learning using neural networks is called *Deep Learning*, if it has a "high" number of hidden layers in order to capture non-linearities (depending on the context "high" means just "> 2")
- "Deep Hedging" is the idea to use deep learning to find the hedging strategy ϕ

Sources and References

Deep Hedging is an active field of research. This presentation is based mostly on the following sources:

- Black/Scholes model: M. Groncki's blog post & notebook (many thanks for allowing us to use it for this talk)

  https://ipythonquant.wordpress.com/2018/06/05/

  option-hedging-with-long-short-term-memory-recurrent-neural-networks-part-i/

- Heston model: Bühler, Gonon, Teichmann, Wood. *Deep Hedging*

  https://arxiv.org/abs/1802.03042

- ETH Lectures (Teichmann)

  https://people.math.ethz.ch/~jteichma/index.php?content=teach_mlf2019

- LSTM Intro (N. Nowaczyk)

  https://github.com/niknow/machine-learning-examples

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Recap: Math Finance Setup

Definition (standard market)

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let

$$dS_t = \mu_t(S_t)dt + \sigma_t(S_t)dW_t$$

be an Ito process of *tradable assets* $S^1, \ldots, S^p$, such that $\mu \in L^1$, $\sigma \in L^2$ (pathwise). Here, W_t is a multi-variate Brownian motion. Let $\mathbb{F} := (\mathcal{F}_t)_{t \geq 0}$ be the augmented natural filtration generated by W_t. We assume that $N := S^1$ is the chosen numeraire and that all deals mature at a maximum maturity $T > 0$.

Self-financing Trading Strategies

Definition (trading strategy)

In a standard market with assets $S^1, \ldots, S^p$, a *trading strategy* is a predictable adapted process $\phi : \Omega \times [0, T] \to \mathrm{R}^p$. The process

$$\Pi(t) := \phi(t)S(t) := \sum_{i=1}^{p} \phi_i(t)S_i(t),$$

is the associated *portfolio value process*. A strategy ϕ is called *self-financing* if

$$d\Pi(t) = \phi(t)dS(t).$$

Definition (arbitrage opportunity)

A trading strategy ϕ is an *arbitrage opportunity*, if its portfolio value process Π satisfies

$$\Pi(0) = 0, \quad \exists 0 \leq t \leq T : \Pi(t) \geq 0 \text{ a.s. and } \mathbb{P}[\Pi(t) > 0] > 0.$$

A market is *arbitrage free*, if there exists no arbitrage opportunities.

Theorem

If there exists an equivalent martingale measure $\mathbb{Q}$, then the market is arbitrage free.

Replication & Completeness

Definition

A *contigent claim* / *derivative* is an $\mathcal{F}_T$-measurable random variable V_T. A trading strategy ϕ *replicates* V_T if its associated value process Π satisfies $\Pi_T = V_T$. A market is *complete* if every claim can be replicated.

Theorem

If there exists a unique equivalent martingale measure $\mathbb{Q}$, then the market is arbitrage free and complete.

The strategy ϕ can be used to hedge the derivative as $\Pi_t = \sum_{i=1}^{p} \phi_t^i S_t^i = V_t$.

Example 1: Black/Scholes model

Definition

The 2-asset market comprising of a deterministic bank account and a GBM for a single stock,

$$dB_t = rB_t dt, \qquad dS_t = \mu S_t dt + \sigma S_t dW_t,$$

is called *Black/Scholes model*.

Theorem

The Black/Scholes model is arbitrage-free and complete. It has the explicit representation

$$B_t = B_0 e^{rt}, \qquad S_t = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}$$

Example 1: Hedging Options in a Black/Scholes model

Theorem

The price of a European call option $C_T = (S_T - K)^+$ with strike K in the Black/Scholes model is given by

$$
\begin{aligned}
C_t &= \Phi(d_1)S_t - \Phi(d_2)e^{-r(T-t)}K \\
d_1 &= (\ln(\tfrac{S_t}{K}) + (r + \tfrac{\sigma^2}{2})(T-t))/(\sigma\sqrt{T-t}), \\
d_2 &= d_1 - \sigma\sqrt{T-t},
\end{aligned}
$$

where Φ is the standard normal cdf. Furthermore,

$$\Delta_t = \frac{\partial C_t}{\partial S_t} = \Phi(d_1).$$

is the amount of stock necessary to replicate C_t.

Example 2: Heston Model Definition

Definition

The *Heston model* for a stock price is defined by

$$dS_t^1 = \sqrt{V_t} S_t dB_t, \qquad S_0^1 = s_0,$$
$$dV_t = \alpha(b - V_t)dt + \sigma\sqrt{V_t}dW_t \qquad , V_0 = v_0,$$

where $\alpha, b, \sigma, s_0, v_0 > 0$ and B_t, W_t are Brownian motions correlated with $\rho \in [-1, 1]$.

The stochastic volatility V_t cannot be traded directly. Thus, we introduce an idealized variance swap on it with price process

$$S_t^2 := \mathbb{E}_{\mathbb{Q}}\Big[\int_0^T V_s ds \mid \mathcal{F}_t^H\Big], \qquad \mathcal{F}_t^H = \sigma((S_s^1, V_s), 0 \leq s \leq t)$$

The variance swap can be written as

$$S_t^2 = \int_0^t V_s ds + L(t, V_t), \quad L(t,v) = \frac{v-b}{\alpha}(1 - e^{-\alpha(T-t)} + b(T-t)).$$

Let $g : \mathbb{R} \to \mathbb{R}$ be the European payoff of an option. By the Markov property, its price process can be written as $C_t = \mathbb{E}_{\mathbb{Q}}[g(S_t^1) \mid \mathcal{F}_t^H] = u(t, S_t^1, V_t)$. We obtain

$$g(S_T^1) = q + \int_0^T \phi_t^1 dS_t^1 + \int_0^T \phi_t^2 dS_t^2.$$

where $q = \mathbb{E}_{\mathbb{Q}}[g(S_T^1)]$, $\phi_t^1 = \partial_s u(t, S_t^1, V_t)$, $\phi_t^2 = \frac{\partial_v u(t, S_t^1, V_t)}{\partial_v L(t, V_t)}$.

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Practical Problem 1: Trading is not continuous

In theory, continuous trading is and a bank can short the option C_t and hedge it by going long the replication portfolio Π with the trading strategy ϕ (for both Black/Scholes and Heston). This is called *Delta Hedging* and would result in a net position of:

$$-C_t + \Pi_t = 0.$$

In reality, continuous trading is not possible and we have to work on a discretized grid $0 = t_0 < \ldots < t_n = T$ leaving us with a *hedge error* at maturity of

$$-C_T + p_0 + \underbrace{\sum_{k=0}^{n-1} \phi_{t_k}(S_{k+1} - S_k)}_{=:(S\cdot\phi)_T},$$

where p_0 is an initial injection of cash into the trading strategy ϕ.

Practical Problem 2: Incomplete Markets & Frictions

In theory, we have assumed that the market is complete and thus every derivative can be perfectly hedged and that all trading is frictionless.

In reality, a market might not be complete and all trading comes with friction such as transaction costs c_k. Thus, this leaves us with a total portfolio value of

$$PL_T(Z, p_0, \phi) := -C_T + p_0 + (S \cdot \phi)_T - \underbrace{\sum_{k=0}^{n} c_k(\phi_{t_k} - \phi_{t_{k-1}})}_{=: C_T(\phi)},$$

and we can no longer assume that this will balance out to zero, but we can minimize this random variable in some metric!

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Convex Risk Measure Definition

Definition

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and let $\mathcal{X}$ be the space of all real valued random variables on Ω. Then $\rho : \mathcal{X} \to \mathrm{R}$ is a *convex risk measure*, if for any X, X_1, X_2 (representing portfolio assets):

1. Monotone decreasing: $X_1 \geq X_2 \Longrightarrow \rho(X_1) \leq \rho(X_2)$.
2. Convex:
   $\forall \alpha \in [0,1] : \rho(\alpha X_1 + (1-\alpha)X_2) \leq \alpha\rho(X_1) + (1-\alpha)\rho(X_2)$.
3. Cash-Invariant: $\forall c \in \mathrm{R} : \rho(X + c) = \rho(X) - c$.

The measure ρ is called *normalized* if $\rho(0) = 0$.

Convex Risk Measures and Hedge Error

Theorem

Let ρ be a convex risk measure and $\mathcal{H}$ be the space of all constrained trading strategies. If $C_T(\cdot)$ and $\mathcal{H}$ are convex, then

$$X \mapsto \pi(X) := \inf_{\phi \in \mathcal{H}} \rho(X + (\phi \cdot S)_T - C_T(\phi))$$

is a convex risk measure as well.

Definition

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. For any $\alpha \in [0,1[$, the risk measure

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

is called *expected shortfall with risk-aversion level* α. Here,

$$\mathrm{VaR}_\gamma(X) := \inf\{m \in \mathbf{R} : \mathbb{P}(X < -m) \leq \gamma\}$$

is the *Value-at-risk with confidence level* γ.

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Hedging Problem

Let $S = (S_1, \ldots, S_p)$ be tradable assets in a market and X be the payout of a derivative. We assume that $0 = t_0 < \ldots < t_n = T$ are the distinct points at which trading is possible. We want to find a trading strategy $\phi = (\phi_0, \ldots, \phi_n)$, $\phi_j \in \mathbf{R}^k$, such that ϕ attains

$$\pi(X) = \inf_{\phi \in \mathcal{H}} \rho(X + (\phi \cdot S)_T - C_T(\phi))$$

for some convex risk measure ρ.
Idea: Generate realizations of S and "learn" the trading strategy ϕ at each point in time t_k.

Problem: ϕ_t depends on t!

Outline

1 Introduction
- Recap of Math Finance Setup & Notation
- Hedging: Theory vs. Practice
- Risk Measures
- General Hedging Problem

2 Recurrent Neural Networks (RNN)
- Predicting Sequences
- Long-Term-Short-Term Networks (LSTM)

3 Deep Hedging
- Black/Scholes Numerical Results
- Heston Results

Outline

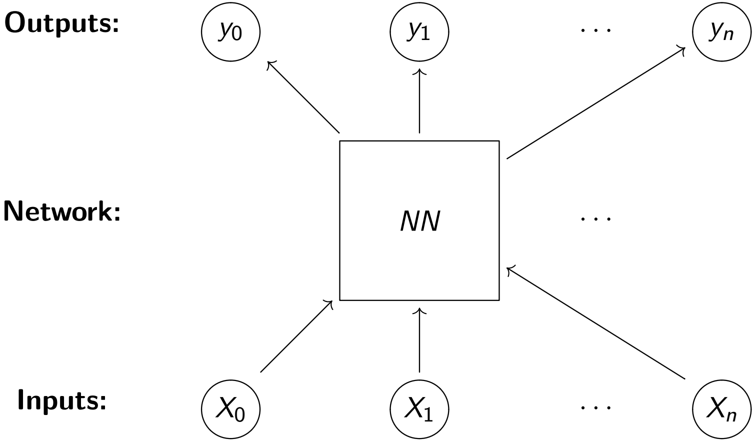
1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Problem context

Usually, an artificial neural network (ANN) needs an input X to predict an output y. How to use ANNs if the input is a sequence X_t and the output is a sequence y_t?

1. Use a big NN with either fixed length vector input/outputs $X_{t_1}, \ldots, X_{t_n}$ and $y_{t_1}, \ldots, y_{t_n}$ or input/output pairs (X, t), (y, t).
2. Use a separate NN for each point t_k.
3. Use a RNN!

Idea 1: Train one huge network



Idea 1: Pros and Cons

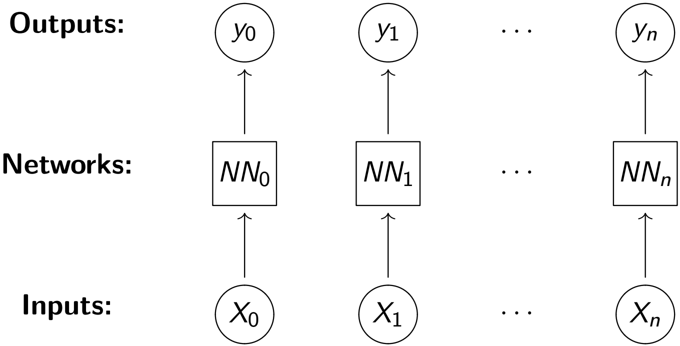
Pros

- easy to understand
- straight-forward to implement

Cons

- resulting NN might need to be huge
- training set required to train the network might be huge
- resulting computational power required might be huge

Idea 2: Train one Network for each element



Idea 2: Pros and Cons

Pros

- resulting NN can be much smaller
- still easy to understand and straight-forward to implement

Cons

- the NN in later positions of the sequences might struggle to make accurate predictions
- predictions are made by independent NN, thus the output might suffer from consistency issues

Origin of this approach: NLP

- Assume we have a text and the quantity to predict is the last word in the sentence. If the second sentence is: *Today is Tuesday*, this can be quite hard. But if the first sentence is: *Yesterday was Monday*, then this becomes a lot easier! Thus a chain of networks can be much more efficient if information is transported through time. (Short Term Memory)
- However, if the $t = 12$ sentence is *Roses are red*, then the information that the predicted word at $t = 1$ was *Tuesday* is not helpful and might actually be harmful. (Long Term Memory)
- Making the prediction at t depend on everything before, blows up the size of the networks. Thus, one needs an efficient way to manage short term and long term memory in these networks to make accurate predictions!

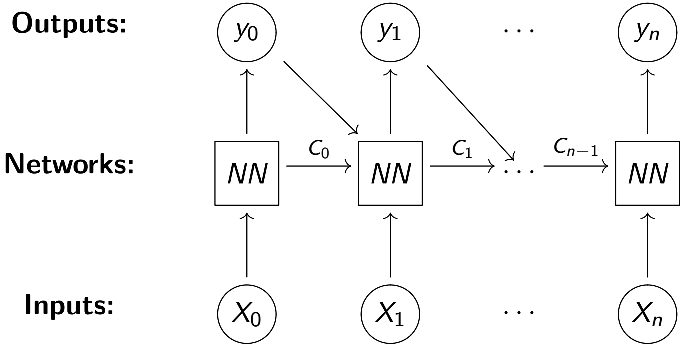
Outline

1 Introduction
- Recap of Math Finance Setup & Notation
- Hedging: Theory vs. Practice
- Risk Measures
- General Hedging Problem

2 Recurrent Neural Networks (RNN)
- Predicting Sequences
- Long-Term-Short-Term Networks (LSTM)

3 Deep Hedging
- Black/Scholes Numerical Results
- Heston Results

Solution: Cell States



Formal Definition of Parameters

Definition (LSTM)

A *Long-Term-Short-Term-Memory neural network* is a tuple $\mathrm{LSTM} = \mathrm{LSTM}(W, U, b, \tau, \sigma)$ consisting of

- a number m of units and a number k of features,
- a 4-tuple W of matrices $W_i, W_f, W_c, W_o \in \mathbb{R}^{k \times m}$ called *input*, *forget*, *cell* and *output kernels*,
- a 4-tuple U of matrices $U_i, U_f, U_c, U_o \in \mathbb{R}^{m \times m}$ called *input*, *forget*, *cell* and *output recurrent kernels*,
- a 4-tuple b of vectors $b_i, b_f, b_c, b_o \in \mathbb{R}^m$ called *input*, *forget*, *cell* and *output bias*,
- two functions $\sigma, \tau : \mathbb{R} \to \mathbb{R}$ called *activation* and *recurrent activation*.

None of the parameters are time-dependent!

LSTM Feedforward

Definition

Let LSTM $=$ LSTM(W, U, b, τ, σ) and $T \in \mathbb{N}$. Any sequence $x = (x_1, \ldots, x_T)$, $x_t \in \mathbb{R}^k$, is called an *input sequence*. We recursively define:

$$\text{input: } i_t := \tau(x_t \odot W_i + h_{t-1} \odot U_i + b_i) \in \mathbb{R}^m, \tag{1}$$

$$\text{forget: } f_t := \tau(x_t \odot W_f + h_{t-1} \odot U_f + b_f) \in \mathbb{R}^m, \tag{2}$$

$$\text{candidate: } \tilde{c}_t := \sigma(x_t \odot W_c + h_{t-1} \odot U_c + b_c) \in \mathbb{R}^m, \tag{3}$$

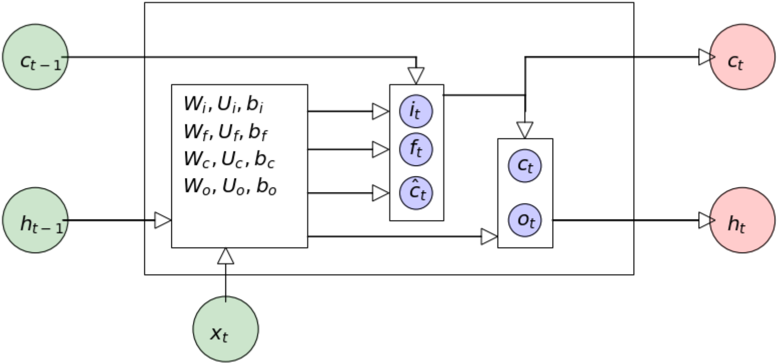
$$\text{output: } o_t := \tau(x_t \odot W_o + h_{t-1} \odot U_o + b_o) \in \mathbb{R}^m, \tag{4}$$

$$\text{cell: } c_t := f_t \bullet c_{t-1} + i_t \bullet \tilde{c}_t \in \mathbb{R}^m, \tag{5}$$

$$\text{carry: } h_t := o_t \tau(c_t) \in \mathbb{R}^m. \tag{6}$$

Conventions: All vectors are row vectors, $c_0 := 0$, $h_0 := 0$, $\odot$ denotes matrix-vector multiplication, $\bullet$ denotes the element-wise multiplication of vectors, σ and τ are applied elementwise. Finally, the function $F_T : \mathbb{R}^{k \times T} \to \mathbb{R}^m$, $x = (x_1, \ldots, x_T) \mapsto h_T$ is called, the *feedforward of* LSTM *of length* T.

LSTM Gate Representation



Remarks

- Recurrent Neural Networks in general and LSTMs in particular exist in many different variants.
- The internal structure of every node is a neural network and in principle, one can play around with any choice of architecture.
- Popular choices are hidden layers or different activation functions.
- The choices made above are in line with the `keras` implementation of `LSTM`.

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Deep Hedging

- *Deep Hedging* is the technique of applying a recurrent neural network, for example the LSTM, to obtain a trading strategy ϕ of portfolio hedging a derivative security by using realizations of the underlying S as inputs.
- By giving the training algorithm of the neural network a risk metric such as ES as the cost function, the neural network will solve the problem of finding the optimal hedging strategy automatically. Thus, this is an instance of unsupervised learning.
- Generating the sample paths of the underlying is a non-trivial finance problem, which needs to be solved by Monte Carlo methods as usual.

Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)
3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

Black/Scholes Setup

- The easiest example is provided by using an LSTM to learn the Δ-hedging strategy of a Call Option in the Black/Scholes model.
- There is a nice blog post and a Jupyter notebook by M. Groncki on this topic (thanks for allowing us to use it in this workshop!):

  `https://ipythonquant.wordpress.com/2018/06/05/`

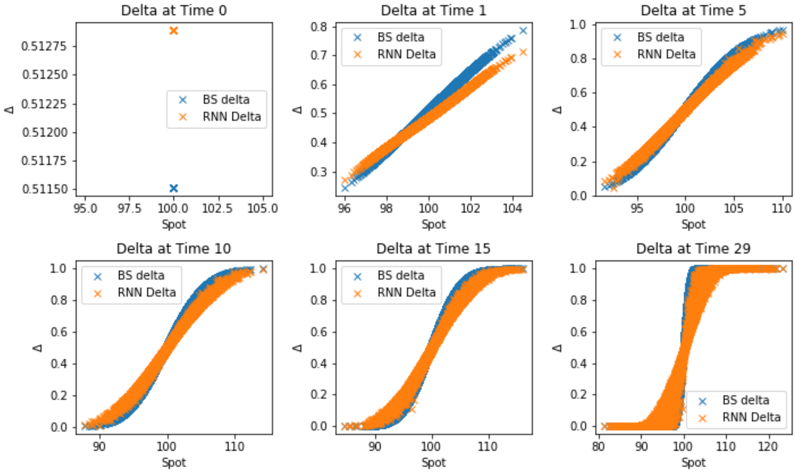
  `option-hedging-with-long-short-term-memory-recurrent-neural-networks-part-i/`

  `https://github.com/mgroncki/DataScienceNotebooks/tree/master/DeepHedging`
- In the following, we summarize a (slightly modified) version of Groncki's results.

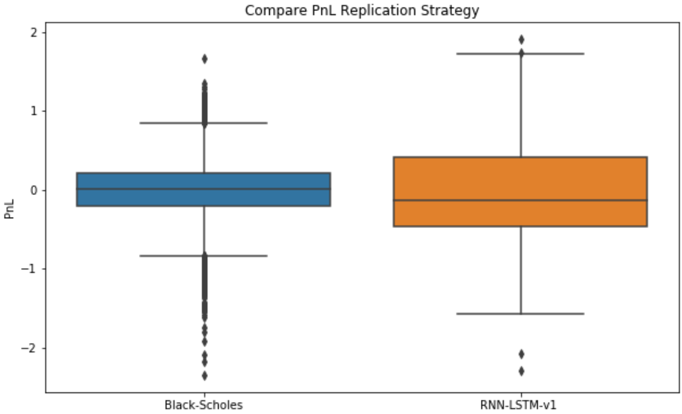
Setup

- We consider the Black/Scholes model S_t with $\mu = 0$, $\sigma = 20\%$ on a daily grid with a $T = 1M$ horizon.
- We train an LSTM on $500k$ realizations on all 30 days.
- We present the discretization error of the Black/Scholes strategy and perform various comparisons between the analytic Black/Scholes-Delta and the strategy obtained by the LSTM.

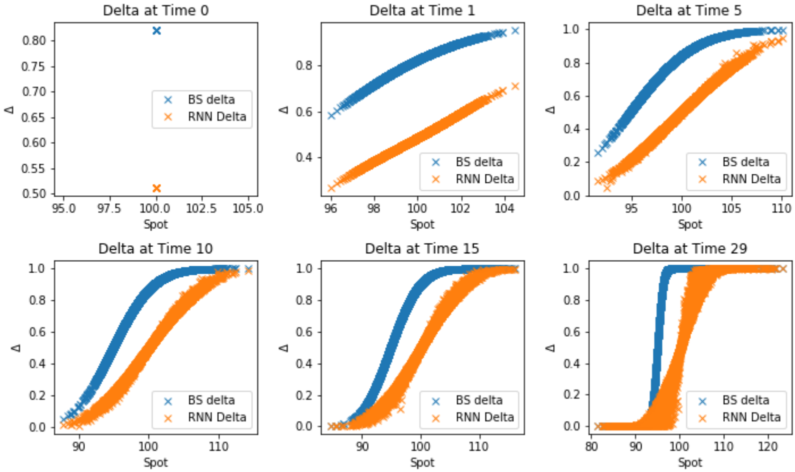
1) Out of Sample Test (same moneyness): Deltas



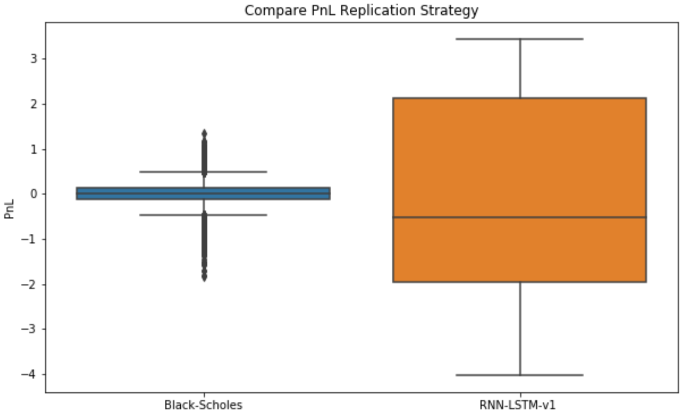
1) Out of Sample Test (same moneyness): Boxplot



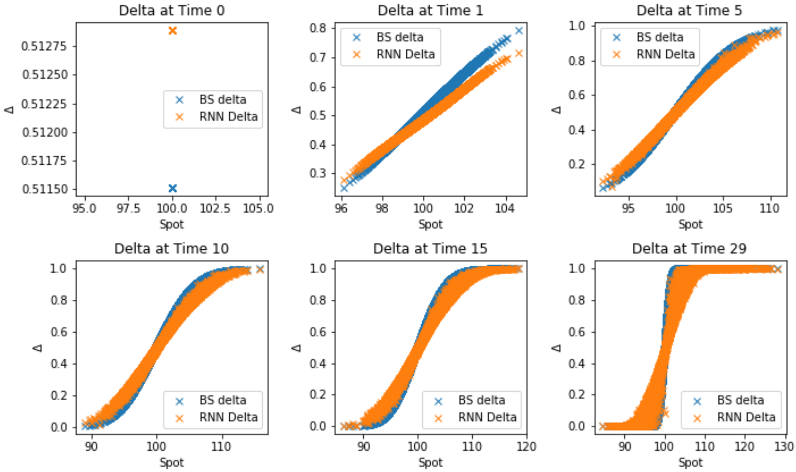
2) Out of Sample Test (different moneyness): Deltas



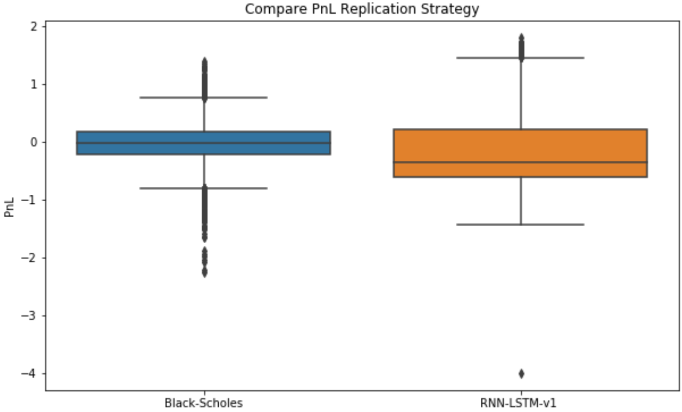
2) Out of Sample Test (different moneyness): Boxplot



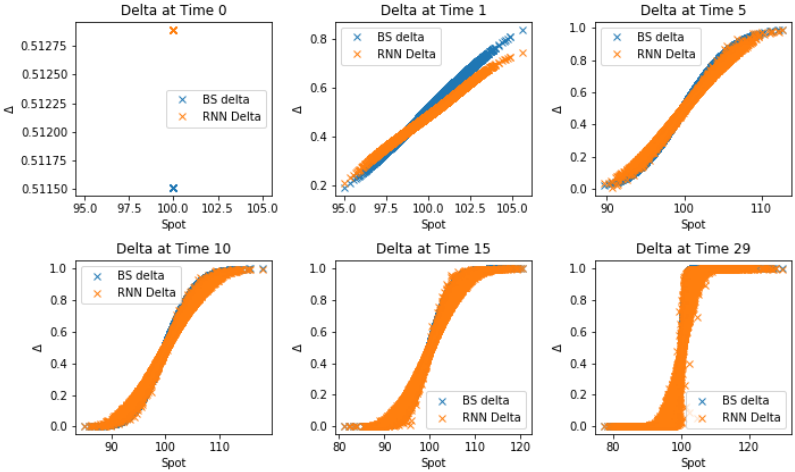
3) Out of Sample Test (different drift): Deltas



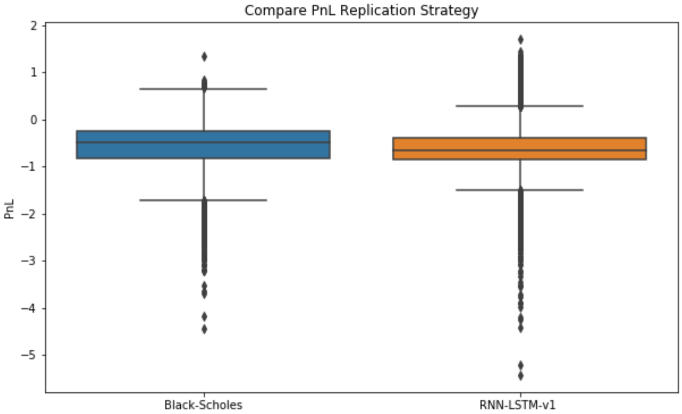
3) Out of Sample Test (different drift): Boxplot



4) Out of Sample Test (different volatility): Deltas



4) Out of Sample Test (different volatility): Boxplot



Outline

1. Introduction
   - Recap of Math Finance Setup & Notation
   - Hedging: Theory vs. Practice
   - Risk Measures
   - General Hedging Problem

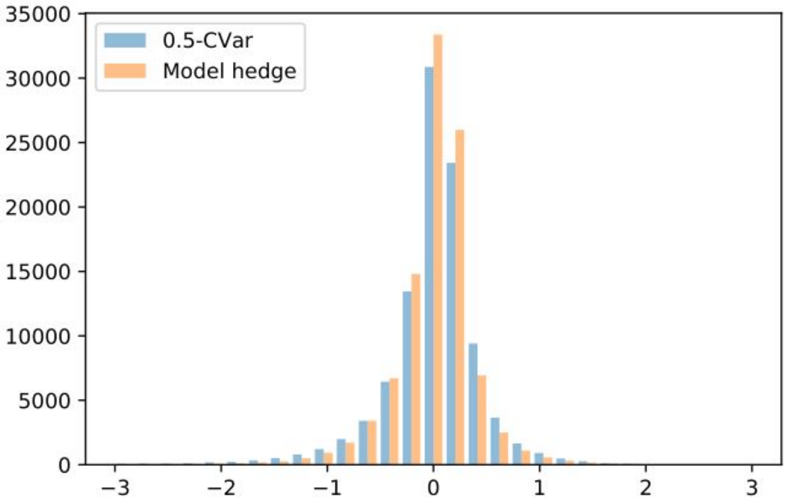
2. Recurrent Neural Networks (RNN)
   - Predicting Sequences
   - Long-Term-Short-Term Networks (LSTM)

3. Deep Hedging
   - Black/Scholes Numerical Results
   - Heston Results

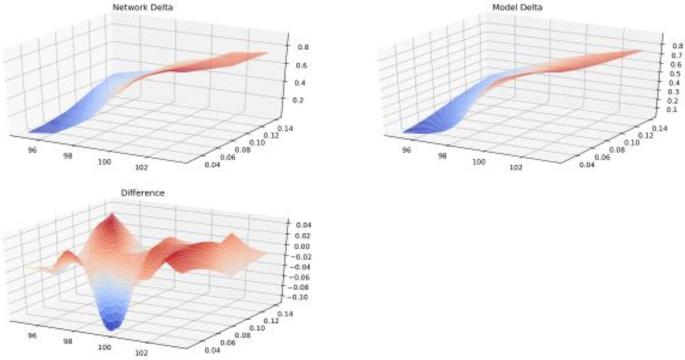
Setup

- We now present the results obtained by Bühler et al. in: https://arxiv.org/abs/1802.03042
- The setup is the discretized Heston model as discussed above and a Call Option as a derivative.

Model Hedge vs Deep Hedge $\alpha = 50\%$



Model Hedge vs Deep Hedge $\alpha = 50\%$ at fixed time slice



Further Research Questions

- The Black/Scholes pricing function has also been studied and can be learned (including the time-dependence) with just one NN:

  `https://arxiv.org/abs/1901.08943v2`

  Given that the Δ is Black/Scholes is "easier" than the pricing, is LSTM really necessary?
- Can Groncki's approach be improved such that one has one NN to price options for all (reasonable) choices of moneyness?
- How to systematically evaluate such an NN approach? What would be needed to pass model validation?
- Can one gain insights into the problem by studying the topology of a NN that is (approximately) optimal in a certain sense?