Data-Driven Models for Discrete Hedging Problem

Ke Nian
Supervisors: Prof.Yuying Li and Prof.Thomas.F.Coleman

David R. Cheriton School of Computer Science,
University of Waterloo,
Waterloo, Canada

January 22, 2019

Agenda

Introduction

Delta Hedging Variants
- Stochastic Volatility Model
- Minimum Variance Approach
- Local Volatility Model

Data Driven Approach
- Indirect Data-Driven Approach
- Direct Data-Driven Approach
- Real Data Experiments

Sequential Learning Framework
- Motivation
- Recurrent Neural Network

Research Focuses and Potential Contributions

Timeline

Practitioner Black-Scholes (BS) Delta Hedging

- BS model:

$$\frac{dS}{S} = \mu dt + \sigma dZ$$

$$\sigma : \text{ Constant}$$

- Implied Volatility

$$\sigma_{imp} = V_{BS}^{-1}(V_{mkt}, .)$$

V_{mkt}: market option price
V_{BS}^{-1} : inverse of BS pricing function

- BS Delta:

$$\delta_{BS} = \frac{\partial V_{BS}}{\partial S}$$

Problem with Black-Scholes Delta

Problem with the traditional Black-Scholes delta:

- Market violates BS assumption
- Dependence of volatility on underlying asset price

Variants of Hedging Strategy:

- Stochastic Volatility Model
- Local Volatility Model
- Minimum Variance Approach
- Indirect Data-Driven Approach
- **Direct Data-Driven Approach**

Stochastic Volatility Model

Stochastic volatility models:

- Heston Model

$$
\begin{aligned}
dS_t &= rS_t dt + \sqrt{\upsilon_t} S_t dW_t \\
d\upsilon_t &= \kappa(\overline{\upsilon} - \upsilon_t) dt + \eta \sqrt{\upsilon_t} dZ_t \\
dZ_t dW_t &= \rho dt
\end{aligned}
$$

- Many stochastic volatility models do not have analytical formula for pricing and hedging function.

Minimum Variance Approach

Considering the the dependence of imply volatility on asset price:

- The Minimum Variance (MV) delta:

$$\delta_{MV} = \frac{\partial V_{BS}}{\partial S} + \frac{\partial V_{BS}}{\partial \sigma_{imp}} \frac{\partial \sigma_{imp}}{\partial S}$$

- The authors [1] propose:

$$\frac{\partial \sigma_{imp}}{\partial S} = \frac{a + b\delta_{BS} + c\delta_{BS}^2}{S\sqrt{T}} \quad (1)$$

a, b and c are the parameter to be fitted using market data.

[1] Hull, J. and White, A., "Optimal delta hedging for options." Journal of Banking and Finance 82 (2017): 180-190.

Local Volatility Model

The local volatility function (LVF) [2]: volatility is a deterministic function of S and t.

$$\delta_{MV} = \frac{\partial V_{BS}}{\partial S} + \frac{\partial V_{BS}}{\partial \sigma_{imp}} \frac{\partial \sigma_{imp}}{\partial S}$$

Local volatility model can also be used to calculate the $\frac{\partial \sigma_{imp}}{\partial S}$.

[2] Coleman, T.F., Kim, Y., Li, Y. and Verma, A., 'Dynamic hedging with a deterministic local volatility function model,' Journal of risk, 4 ,1 (2001):63-89

Problem with Parametric Approach

Parametric approaches:

- Model mis-specification.
- Sub-optimal for discrete hedging problems.

Data-driven approaches:.

- Minimum assumptions on S.
- Model is determined by market data.

Indirect Data-driven Approach

The indirect data-driven approach [3] can be summarized as following:

- Let X be the features from market.
  - Asset price S
  - Strike Price K
  - Time to expiration $T - t$
- Determine the data driven pricing function $V(X)$ using regression model.
- Compute

$$\delta_{ID} = \frac{\partial V(X)}{\partial S}$$

[3] Hutchinson, J.M., Lo, A.W. and Poggio, T., "A nonparametric approach to pricing and hedging derivative securities via learning networks." The Journal of Finance 49.3 (1994): 851-889.

Problem with Indirect Data-Driven Approach

Problem with the Indirect Data-Driven Approach:

- Unnecessary intermediate procedure.
- Sub-optimal for discrete hedging.
- Model parameters depend on the asset price.

Direct data-driven approach can be more useful in practice.

- Customized hedging position function.
- Directly compute the hedging position.

Direct Data-driven Approach

The direct data-driven approach is

$$\min_f \left[\frac{1}{N} \sum_{i=1}^{N} (\Delta V_i - \Delta S_i f(X_i))^2 \right]$$

$$\Delta V_i : \text{ the change of option value in data instance } i$$
$$\Delta S_i : \text{ the change of asset price in data instance } i$$

Real Data Hedging Experiments

- Data: S&P 500 index option from Jan 2007 and Aug 2015
- Model Calibration:
  - SABR: daily calibration
  - LVF: $\frac{\partial \sigma_{imp}}{\partial S}$ from implied volatility surface
  - MV: Use a 36 months time window to train
  - $\mathrm{DKL}_{\mathsf{SPL}}$: Use a 36 months time window to train. Models are separately calibrated for different Black-Sholes delta range.

Evaluation Criteria: Local Risk

The percentage increase in the effectiveness over the BS hedging:

$$Gain = 1 - \frac{SSE[\Delta V_i - \Delta S_i \delta^i]}{SSE[\Delta V_i - \Delta S_i \delta^i_{BS}]}$$

SSE: sum of squared errors

δ: hedging position computed from different models

δ_{BS}: BS delta

S&P 500 Call Options

Delta	SABR (%)	LVF (%)	MV (%)	$\mathrm{DKL}_{\mathsf{SPL}}$ (%) Leave-One-Out[1] Traded	$\mathrm{DKL}_{\mathsf{SPL}}$ (%) Leave-One-Out[1] All
0.1	42.1	39.4	42.6	**44.1**	**44.4**
0.2	35.8	33.4	36.2	**37.8**	**38.1**
0.3	31.1	29.4	30.3	**33.1**	**33.6**
0.4	28.5	26.3	26.7	**30.9**	**31.3**
0.5	27.1	24.9	25.5	**30.0**	**30.4**
0.6	25.7	25.2	25.2	**29.3**	**29.8**
0.7	25.4	24.7	25.8	**28.4**	**30.2**
0.8	24.1	23.5	25.4	22.5	**28.0**
0.9	16.6	**17.0**	16.9	8.1	12.7
Overall	25.7	24.6	25.5	**31.3**	**26.8**

Table: S&P 500 Call Option Daily Hedging: bold entry indicating best Gain

[1] For each month, the penalties for models are determined by leave-one-out cross validation.

Volatility Clustering and Financial Time Series

- Sequential learning framework may further improve the performance:
  - Volatility clustering observed in the financial market.
  - Autocorrelation between data instances near in time.
- Recurrent Neural Networks (RNNs) are popular models that have been widely used in time series analysis.

Recurrent Neural Network (1)

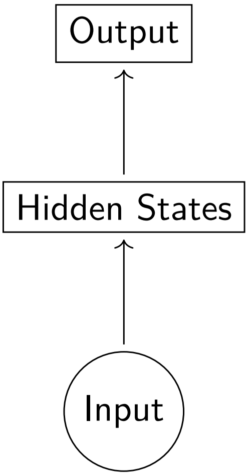


Figure: Neural Network

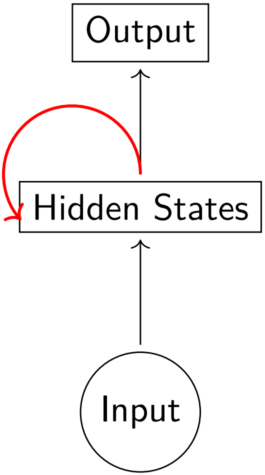
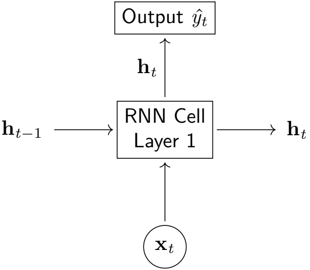


Figure: Recurrent Neural Network

Recurrent Neural Network (2)



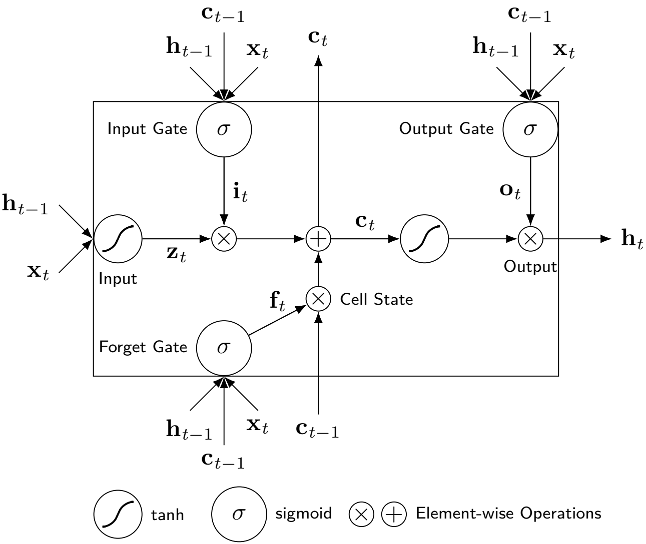
In each RNN cell:

$$\mathbf{h}_t = f_{act}(\boldsymbol{W}_{hx}\mathbf{x}_t + \boldsymbol{W}_{hh}\mathbf{h}_{t-1} + b_h)$$
$$\hat{y}_t = f_{out}(\boldsymbol{W}_{yh}\mathbf{h}_t + b_y)$$

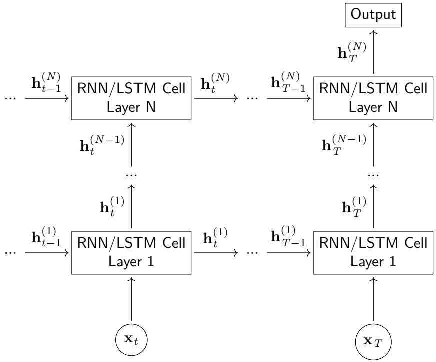
- The original RNN model suffers from the problem of vanishing gradients.
- Gated Recurrent Unit (GRU) [4]model is introduced to combat vanishing gradients through a gating mechanism.

[4]Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

Long Short-Term Memory(LSTM) Model

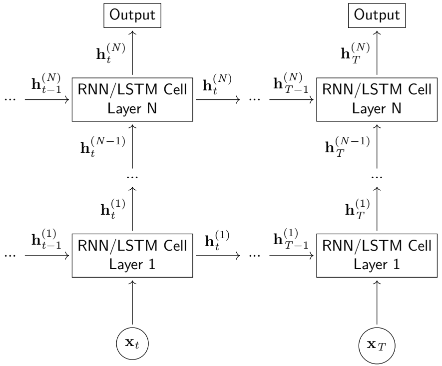


Potential Usage: Many-to-one Model



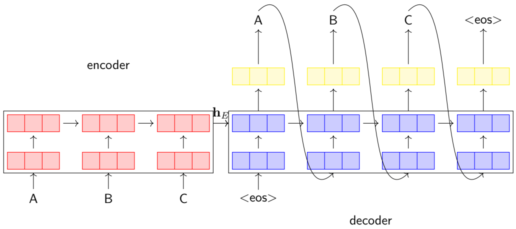
This framework is suitable for one-step discrete hedging problems.

Potential Usage: Many-to-many Model



This framework is suitable for multi-step discrete hedging problems.

Potential Usage: Feature Extraction



Transformation of sequences of variable lengths to fixed size feature vectors h_E.

Benefits of RNN framework

Benefits:

- Ability to cope with sequences of variable lengths.
- Ability to learn dependence of data near in time.
- Online learning.

David R. Cheriton School of Computer Science, University of Waterloo

Exploration based on RNN framework

- Investigating the effectiveness for RNN/LSTM framework on discrete hedging problems:
  - Effectiveness on one-step hedging problems.
  - Effectiveness on multi-step hedging problems.
- Feature selection and feature extraction:
  - Identify important features for discrete hedging problems.
  - Extract useful features from time series data.

Challenge: Non-convex Problem

Computation issues related non-convexity:

- 1st-order method vs 2nd-order method
  - LSTM has more parameters and tends to over-fit the data.
  - Simple RNN with 2nd-order method may perform better.
- Weight initialization
- Model pre-training

Challenge: Regularization

The RNN/LSTM models tend to over-fit the data and they are hard to be regularized. We plan to investigate approaches to alleviate the over-fitting problems.

- L1/L2 regularization of the weight matrices.
- Data augmentation.

Comparison of Different Methods

Comparison of different machine learning frameworks on the discrete hedging problems:

- **Robustness**: The learning framework should be robust to the existence of the market crashes.
- **Efficient Computation**: The learning framework should be computational efficient so that large scale training is possible.
- **Online Learning**: The learning framework should be able to incorporate the data whenever it is observed in the market to quickly adjust itself to the market changes.

Potential Contributions

- Identify the drawbacks of classical parametric hedging models.
- Propose and demonstrate the effectiveness of data-driven hedging models based on state-of-the-art machine learning frameworks.
- Identify important features for discrete hedging problem and provide feature extraction framework for the learning process.

Timeline and Progress

Timeline	Work	Progress
	Data-driven Kernel Model	Completed [5]
Winter 2018	Exploration of RNN Framework	In Progress
Spring 2018	Exploration of RNN Framework	In Progress
Fall 2018	Exploration of RNN Framework	In Progress
Winter 2019	Comparison of Different Models	In Progress
Spring 2019	Thesis Writing	In Progress
Fall 2019	Thesis Defense	In Progress

[5] Nian,K., Coleman,T.F. and Li, Y. "Learning Minimum Variance Discrete Hedging Directly from Market.", Quantitative Finance

Q & A

Thank you very much!

Any Questions?