Enhancing Delta Hedging with Neural Networks

Mark Bogorad, Joaquin Garay, Matthew Rubenstein, Thomas Shen

FRE7773 – Machine Learning in Financial Engineering
Fall 2024

What is Dynamic hedging?

What is it to hedge a position?

Static vs. Dynamic Hedging?

Why Hedge Dynamically?

Dynamically Hedging Options

- Black Scholes Framework

$$C = S_0 e^{-qT}\Phi(d_1) - Ke^{-rT}\Phi(d_2)$$

$$P = -S_0 e^{-qT}\Phi(-d_1) + Ke^{-rT}\Phi(-d_2)$$

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}$$

- The Delta Hedge
  - **Delta:** The sensitivity of an option price to movements in the underlying asset price
  - Uniqueness of the delta hedge (compared to hedging with the other Greeks)

Delta-Hedging Options without Machine Learning

- Introduction of the Black-Scholes model in 1973
  - Delta-hedges, higher order hedges

- Expanding the Black Scholes:
  - Stochastic volatility models (e.g. constant elasticity)
  - Jump-Diffusion
  - Nonlinear Least Squares

Limitations of traditional approaches

- Continuous rebalancing incurs infinite transaction costs: Time Discretization:
  - Introduces truncation error
- Hedging with multiple options (ex: Delta-Vega hedging)
  - Operational constraints (low liquidity and high slippage)
- Complex models
  - introduces inconsistent pricing bias across time
  - NLS: Relies on iterative algorithms that may not scale well
  - The machine learning model we introduce works similarly to NLS
    - Scales better to high-dimensional data when applying regularization techniques

The Evolution of Dynamic-Hedging

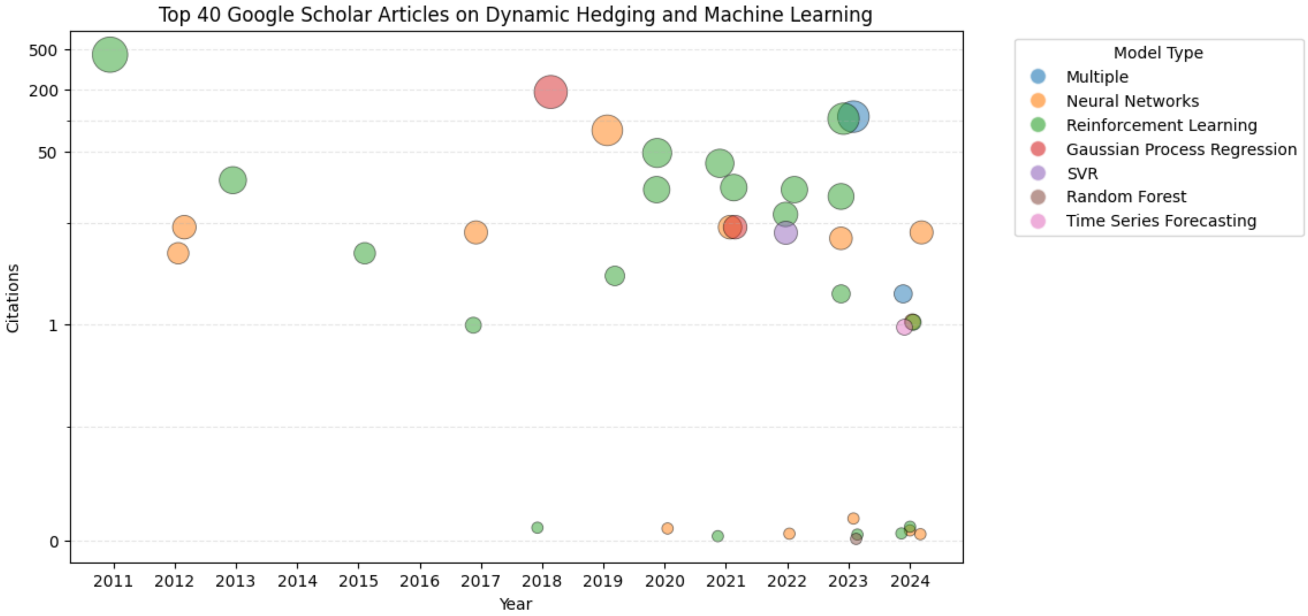
Beginning of Hedging
- Static methods with Black-Scholes framework (1973)

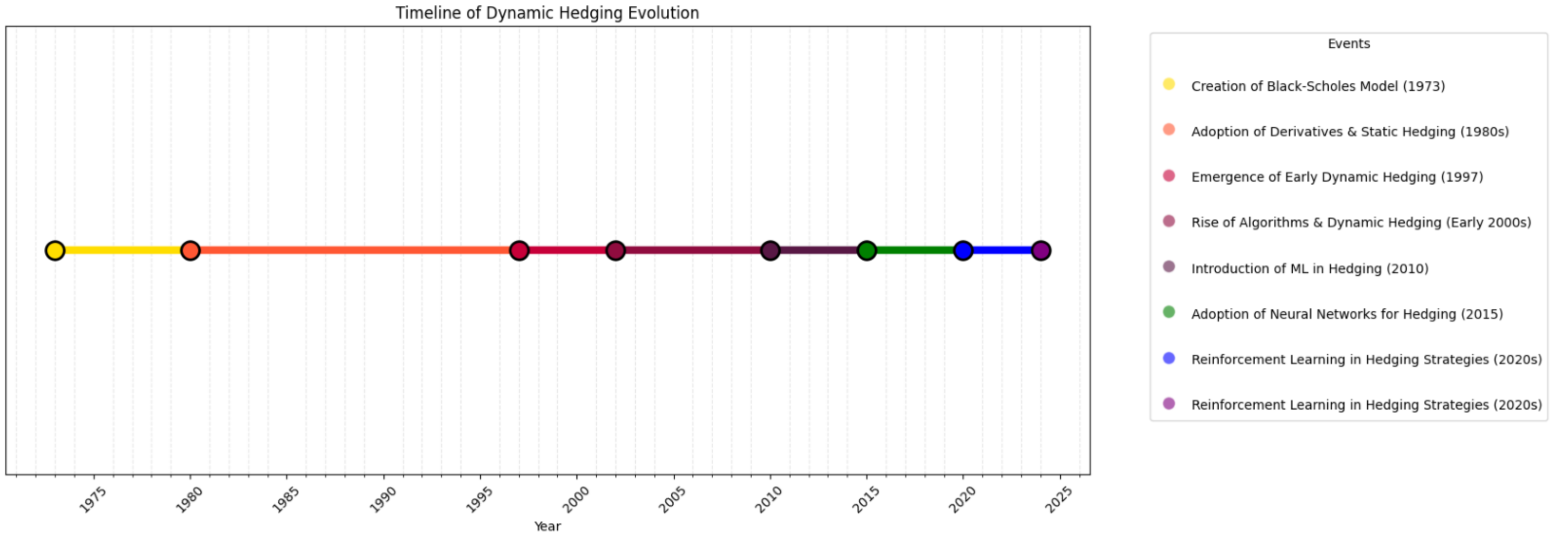
Pre-Machine Learning Methods (1990s)
- Classical quant suite of derivative pricing used for hedging (Monte-Carlo, FDM, binomial framework)

Change to Financial Landscape: SEC (2007)
- Hedging can now happen at much higher frequency
- Birth of modern data collection – a precursor for ML metho

Enter Early Machine Learning Methods (2010)
- SVR - handle non-linear relationships (useful for implied volatility)
- Decision Trees & Random Forests – segment data into volatility regimes & reduce noise

Most Recent Decade: Deep Learning (2015+)
- Neural Networks (2015 onwards)
- Reinforcement Learning (2020 onwards)





NYU

Choice of Application

- Potential Applications
  - Operational Decisions: when to rebalance?
  - Analytical Judgments: what is the best hedging ratio?

- Big Consideration: what kind of data do we have?

Because of data availability, we are going to tackle the problem of predicting **optimal delta hedge ratios**.

Choice of Model

- Most Popular Approaches: Neural Networks vs. Reinforcement Learning
  - Neural Networks are better for predicting hedge ratios

- Why neural networks over less popular regression approaches?
  - SVR
  - Decision Trees

Using Neural Networks to Improve Delta-Hedges

- "*Enhancing Black-Scholes Delta Hedging via Deep Learning*" by Chunhui Qiao and Xiangwei Wan

- Setup
  - Gather data on SPX call products at time steps t_i
  - For each product at each time step, calculate empirically optimal hedge using the next day of data

$$\delta_{\text{optimal, t}} = \frac{\Delta C_{t+1}}{\Delta S_{t+1}} = \frac{C_{t+1} - C_t}{S_{t+1} \cdot e^{-q \cdot \Delta t} - S_t}$$

  - Target the residual

$$target_i = \delta^{(\mathrm{i})}_{\text{optimal}} - \delta^{(\mathrm{i})}_{BS}$$

NYU

Using Neural Networks to Improve Delta-Hedges:

Overcoming Traditional Limitations

- Neural Network predictions as a way to manage time-discretization
- Fitting predictions to real market dynamics (avoiding market assumptions)

NYU

Hands-On: Data

Raw data from Bloomberg Terminal:

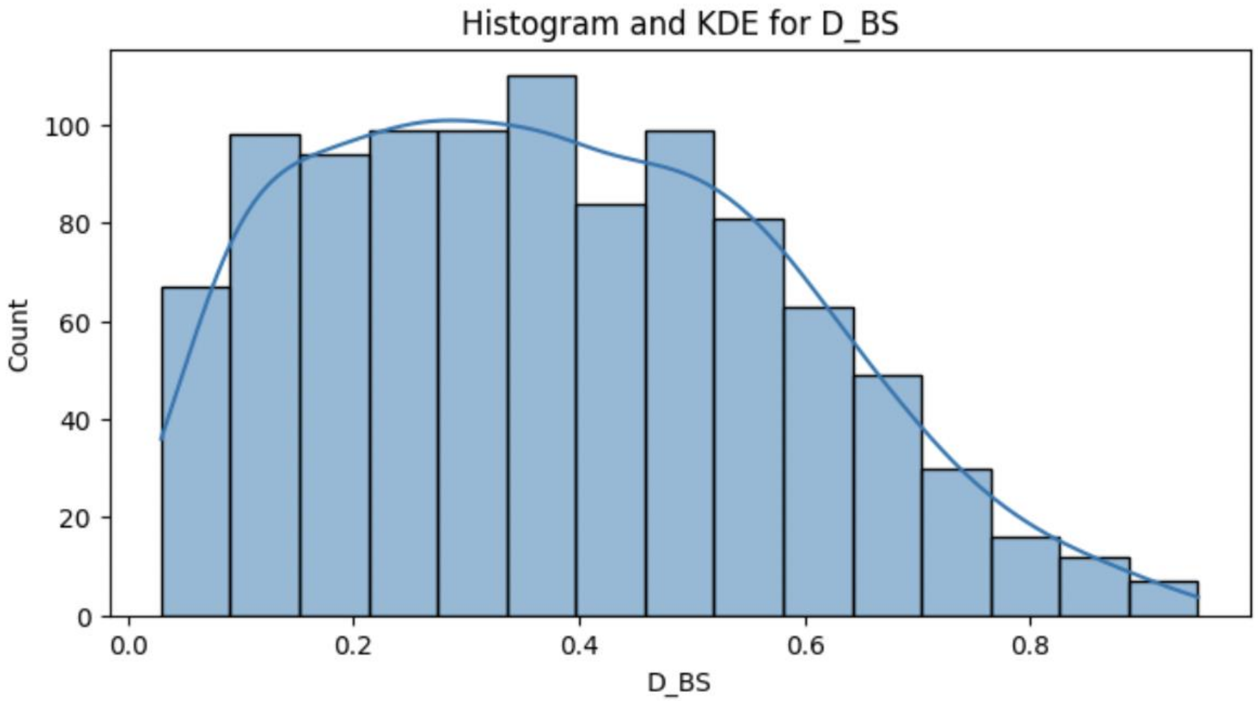
- SPX Index Price,
- SPX Dividend Yield
- Risk Free Rate for each Maturity Date.
- Option Mid Quote
- Option Implied Volatility (based on mid price)
- Option Delta (based on mid price)

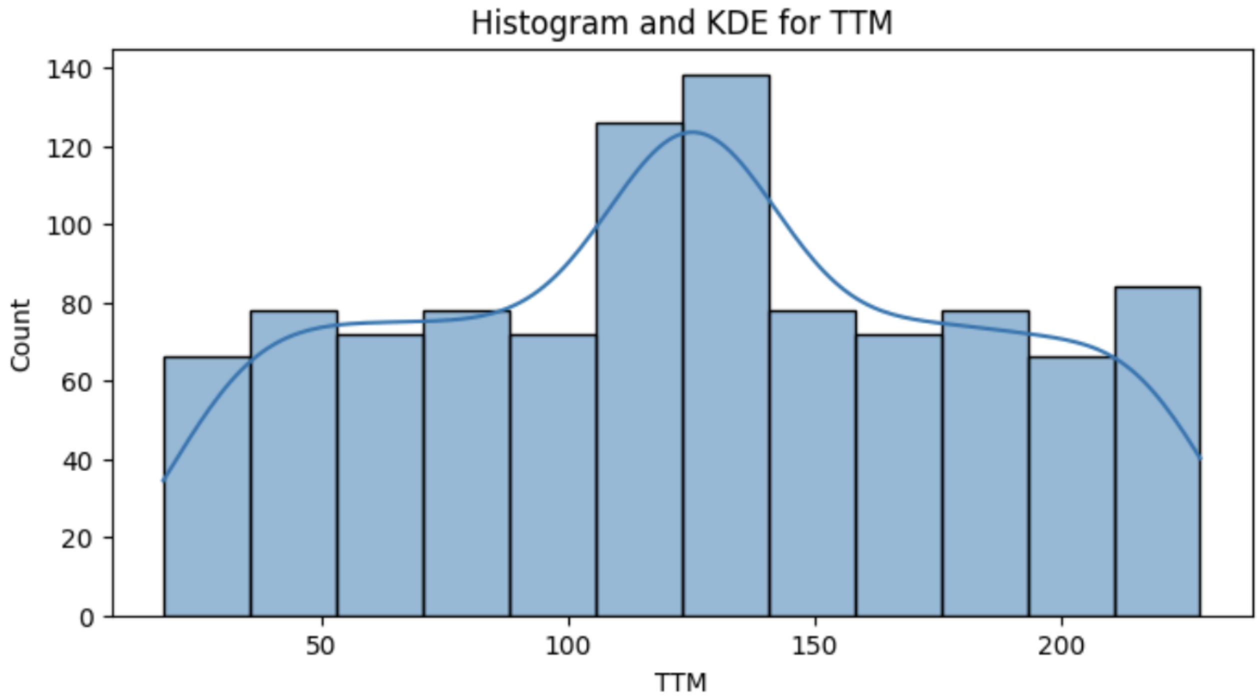
Other Computed Values

- BSM Option Price
- BSM Delta
- Price Changes (Calls and Underlying)
- Optimal (Empirical) Option Delta
- Option Time to Maturity
- Option Moneyness.

ID	Ticker	Type	Strike Price	Maturity Date	Underlying
1	SPX US 3/21/25 C5700 Index	Call	5700	2025-03-21	SPX
2	SPX US 3/21/25 C5800 Index	Call	5800	2025-03-21	SPX
3	SPX US 3/21/25 C5900 Index	Call	5900	2025-03-21	SPX
4	SPX US 3/21/25 C6000 Index	Call	6000	2025-03-21	SPX
5	SPX US 3/21/25 C6100 Index	Call	6100	2025-03-21	SPX
6	SPX US 3/21/25 C6200 Index	Call	6200	2025-03-21	SPX
7	SPX US 12/20/24 C5700 Index	Call	5700	2024-01-20	SPX
8	SPX US 12/20/24 C5800 Index	Call	5800	2024-01-20	SPX
9	SPX US 12/20/24 C5900 Index	Call	5900	2024-01-20	SPX
10	SPX US 12/20/24 C6000 Index	Call	6000	2024-01-20	SPX
11	SPX US 12/20/24 C6100 Index	Call	6100	2024-01-20	SPX
12	SPX US 12/20/24 C6200 Index	Call	6200	2024-01-20	SPX

Example Feature Distributions over Dataset

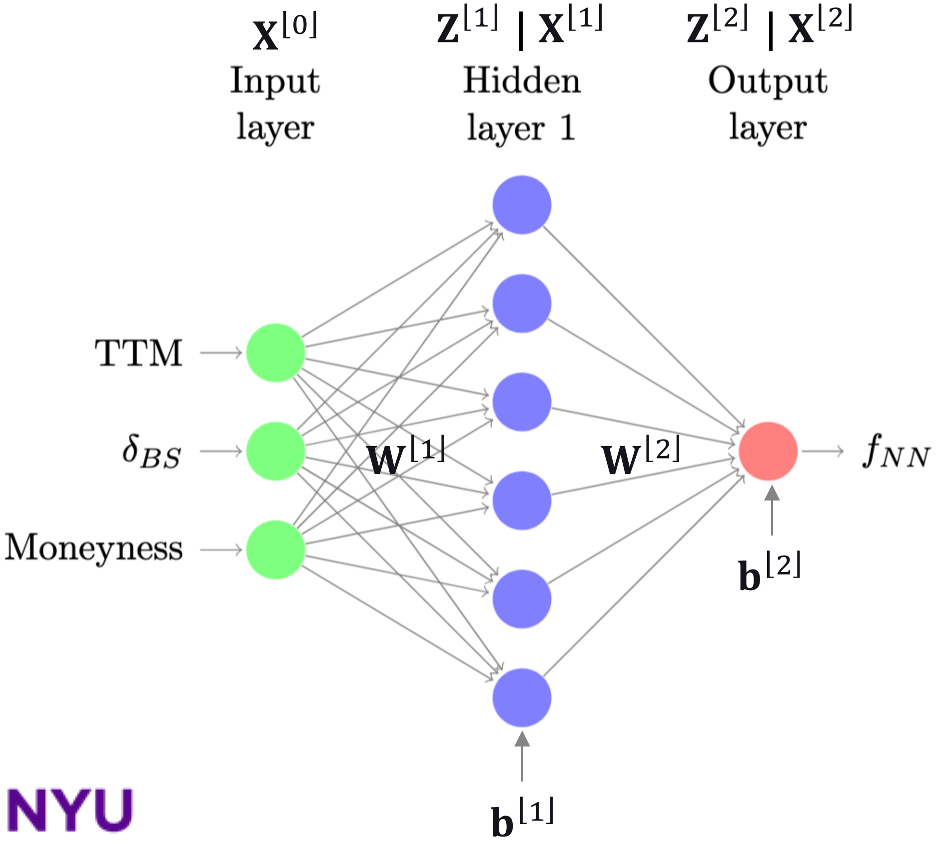




NYU

Hands-On: Setup and Forward Pass

- 3 features: Time-to-maturity, Implied BSM Delta, and Moneyness
- 1-hidden-layer with bias and Sigmoid activation
- MSE Loss function



$$\mathbf{X}^{[1]} = \sigma\big(\underbrace{\mathbf{X}^{[0]}\mathbf{W}^{[1]} + 1_m \mathbf{b}^{[1]}}_{\mathbf{Z}^{[1]}}\big)$$

$$f_{NN} = \mathbf{X}^{[2]} = \mathbf{X}^{[1]}\mathbf{W}^{[2]} + 1_m \mathbf{b}^{[2]}$$

Input-Output Mapping

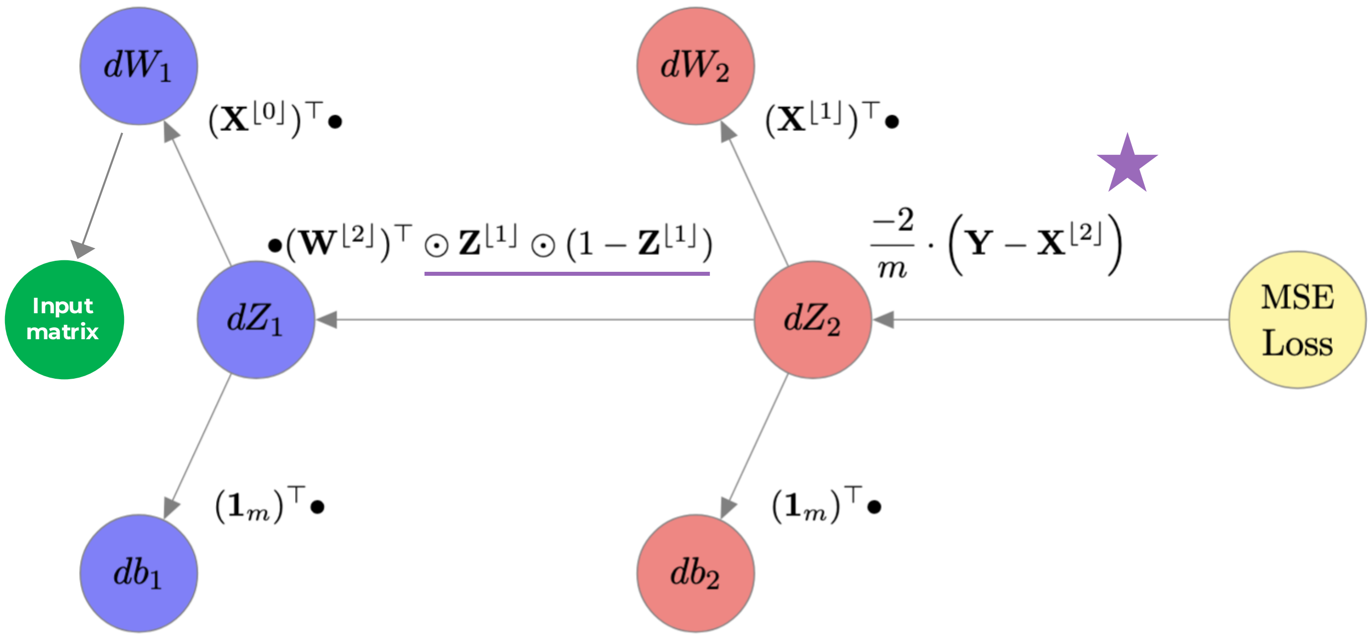
$$\mathcal{L} = \frac{1}{m}\sum_i [y_i - f_{NN}(x_i)]^2 = \frac{1}{m}\left(\mathbf{Y} - \mathbf{X}^{[2]}\right)^\top \left(\mathbf{Y} - \mathbf{X}^{[2]}\right)$$

$$y_i = \delta^{(i)}_{\mathrm{optimal}} - \delta^{(i)}_{BS}$$

MSE Loss Function

NYU

Hands-On: Back Propagation



MLP Back Prop Algorithm

$$\mathbf{X}^{\lfloor 1 \rfloor} = \sigma\left(\mathbf{X}^{\lfloor 0 \rfloor}\mathbf{W}^{\lfloor 1 \rfloor} + 1_m \mathbf{b}^{\lfloor 1 \rfloor}\right)$$

$$f_{NN} = \mathbf{X}^{\lfloor 2 \rfloor} = \mathbf{X}^{\lfloor 1 \rfloor}\mathbf{W}^{\lfloor 2 \rfloor} + 1_m \mathbf{b}^{\lfloor 2 \rfloor}$$

Input-Output Mapping

$$\mathcal{L} = \frac{1}{m}\sum_i [y_i - f_{NN}(x_i)]^2 = \frac{1}{m}\left(\mathbf{Y} - \mathbf{X}^{\lfloor 2 \rfloor}\right)^\top \left(\mathbf{Y} - \mathbf{X}^{\lfloor 2 \rfloor}\right)$$

MSE Loss Function

$$\mathbf{W_{i+1}}^{\lfloor 1 \rfloor} = \mathbf{W_i}^{\lfloor 1 \rfloor} - \alpha \cdot \nabla \mathbf{W_i}^{\lfloor 1 \rfloor}$$

Gradient Descent

For instance,

$$\mathbf{dW}^{\lfloor 1 \rfloor} = (\mathbf{X}^{\lfloor 0 \rfloor})^\top \cdot \frac{-2}{m}(\mathbf{Y} - \mathbf{X}^{\lfloor 2 \rfloor}) \cdot (\mathbf{W}^{\lfloor 2 \rfloor})^\top \odot \mathbf{Z}^{\lfloor 1 \rfloor} \odot (1 - \mathbf{Z}^{\lfloor 1 \rfloor})$$

NYU

Hands-On: Manually solving FNN

- Select 30 samples for training with no validation subset.
- Initialized the values of $\mathbf{W}^{[1]}$, $\mathbf{b}^{[1]}$, $\mathbf{W}^{[2]}$, $\mathbf{b}^{[2]}$ randomly from $\mathcal{U}(-\sqrt{k}, k)$ where $k = \frac{1}{Features\ In}$
- Set the learning rate $\alpha = 0.1$.

Iteration step

$$\mathbf{W}_0{}^{[1]} = \begin{bmatrix} 0.441 & 0.479 & -0.135 \\ 0.530 & -0.126 & 0.117 \\ -0.281 & 0.339 & 0.509 \\ -0.424 & 0.502 & 0.108 \\ 0.427 & 0.078 & 0.278 \\ -0.082 & 0.445 & 0.085 \end{bmatrix}^{\mathsf{T}}, \mathbf{b}_0{}^{[1]} = \begin{bmatrix} -0.270 \\ 0.147 \\ -0.266 \\ -0.068 \\ -0.234 \\ 0.383 \end{bmatrix}^{\mathsf{T}}, \mathbf{W}_0{}^{[2]} = \begin{bmatrix} -0.322 \\ -0.188 \\ -0.115 \\ -0.245 \\ 0.039 \\ -0.403 \end{bmatrix}, \mathbf{b}_0{}^{[2]} = [0.369]$$

$$\mathbf{W}_1{}^{[1]} = \begin{bmatrix} 0.444 & 0.484 & -0.132 \\ 0.532 & -0.124 & 0.118 \\ -0.280 & 0.341 & 0.510 \\ -0.421 & 0.506 & 0.110 \\ 0.426 & 0.078 & 0.278 \\ -0.078 & 0.450 & 0.089 \end{bmatrix}^{\mathsf{T}}, \mathbf{b}_1{}^{[1]} = \begin{bmatrix} -0.266 \\ 0.149 \\ -0.265 \\ -0.065 \\ -0.235 \\ 0.387 \end{bmatrix}^{\mathsf{T}}, \mathbf{W}_1{}^{[2]} = \begin{bmatrix} -0.354 \\ -0.219 \\ -0.144 \\ -0.274 \\ 0.009 \\ -0.438 \end{bmatrix}, \mathbf{b}_1{}^{[2]} = [0.320]$$

$$\boxed{Loss_0 = 13.5854 \rightarrow Loss_1 = 13.5274 \rightarrow Loss_2 = 13.5166}$$

NYU

Code Verification

```
0.441,  0.479,  -0.135,
0.530,  -0.126,  0.117,
-0.281,  0.339,  0.509,
-0.424,  0.502,  0.108,
0.427,  0.078,  0.278,
-0.082,  0.445,  0.085,
hidden.weight: None
---------------------
-0.270,  0.147,  -0.266,  -0.068,  -0.234,  0.383,
hidden.bias: None
---------------------
-0.322,  -0.188,  -0.115,  -0.245,  0.039,  -0.403,
output.weight: None
---------------------
0.369,
output.bias: None
---------------------
Epoch [1/2], Training Loss: 13.5854
```

```
0.444,  0.484,  -0.132,
0.532,  -0.124,  0.118,
-0.280,  0.341,  0.510,
-0.421,  0.506,  0.110,
0.426,  0.078,  0.278,
-0.078,  0.450,  0.089,
hidden.weight: None
---------------------
-0.266,  0.149,  -0.265,  -0.065,  -0.235,  0.387,
hidden.bias: None
---------------------
-0.354,  -0.219,  -0.144,  -0.274,  0.009,  -0.438,
output.weight: None
---------------------
0.320,
output.bias: None
---------------------
Epoch [2/2], Training Loss: 13.5274
```

Extensions and Further Discussion

- Our toy model did not outperform the vanilla BS model delta
  - Data:
    - Qiao and Wan used 10 years' worth of data (2.073 million unique observations)
    - We used data spanning 4 months (1,008 unique observations)
      - Neural-networks are notorious for needing large amounts of data
      - Robustness (market)
  - Parameters:
    - We could include a more comprehensive set of features
  - Loss function:
    - MSE of residual vs MSE hedging error
  - Training methods:
    - Qiao and Wan used Xavier initialization and gradient clipping

```
No batch MSE: 16.7872
30 batch MSE: 16.7455
30 batch norm MSE: 16.7078
Standard greek MSE: 10.51231005280044
```

NYU

Conclusions

Dynamic hedging has grown in tandem with theoretical and technological advancements, from Black-Scholes to deep learning.

Limitations not fully addressed by deep learning methods:

- Dependency on massive datasets
- Decreasing interpretability: very important for a regulatory standpoint
- Even at the deep learning level, we are still subject to human discretion

Hands On – The same story but in finance context

- In its most rudimentary form, we saw that advanced dynamic hedging strategies using modern libraries are the same mechanics just applied to new topics.

The future question – the tradeoff between interpretability and accuracy

NYU

Thank You

Enhancing Delta Hedging with Neural Networks

Fall 2024