On the Limitations of Neural Networks for Option Pricing: Analysis of Volatility Regime Sensitivity

Tarun Raheja*
Kipo AI
tarun@kipo.ai

Nilay Pochhi*
Independent Researcher
pochhi.nilay@gmail.com

Abstract

Recent work demonstrates neural networks' theoretical ability to approximate option pricing functions, but empirical evidence regarding robustness to market regime shifts remains limited. Motivated by practical scenarios where the classical deterministic Black-Scholes equation becomes computationally challenging in high-dimensional settings or under complex market conditions, we examine neural network performance during volatility regime transitions. Models trained on low-volatility regimes ($\sigma = 0.2$) show significant errors under higher volatility ($\sigma = 0.3$). We provide detailed theoretical and empirical analyses indicating that these errors reflect fundamental representational limits of current architectures rather than optimization issues.

1 Introduction

The classical Black-Scholes formula for option pricing is given by:

$$C(S,t) = S\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2), \tag{1}$$

where

$$d_1 = \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = d_1 - \sigma\sqrt{T-t}. \tag{2}$$

Here, $C(S,t)$ is the call option price, S the asset price, K the strike price, r the risk-free rate, T the maturity, and σ the volatility. $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution.

Recent studies suggest neural networks as potential approximators for option pricing Han et al. (2018); Sirignano & Cont (2019), particularly valuable in scenarios involving high-dimensional data or stochastic volatility, where traditional deterministic models become computationally impractical Peng & Yan (2021). Nevertheless, neural network behavior under volatility regime shifts lacks comprehensive analysis.

2 Problem Formulation

Consider a neural network $f_\theta : \mathbb{R}^2 \to \mathbb{R}$ trained to approximate the Black-Scholes pricing function. The network receives inputs (S,t) and outputs an estimated option price. During training, all data is generated with fixed volatility $\sigma_{\text{train}} = 0.2$. At test time, the model encounters data with $\sigma_{\text{test}} = 0.3$, constituting a structured covariate shift.

3 Theoretical Analysis

3.1 Approximation Theory Framework

Let $\mathcal{F}_\theta$ denote the class of feedforward neural networks with ReLU activations. For any $f_\theta \in \mathcal{F}_\theta$, the network partitions the input space into polyhedral regions where the function is locally affine Bartlett et al. (2019). Formally, there exists a set of regions $\{R_i\}_{i=1}^k$ such that:

$$f_\theta(x) = W_i x + b_i \quad \text{for } x \in R_i \tag{3}$$

The Black-Scholes pricing function, however, exhibits distinct nonlinear scaling with volatility:

$$\frac{\partial C}{\partial \sigma} = S\sqrt{T-t}\,\phi(d_1) > 0 \tag{4}$$

where $\phi(\cdot)$ is the standard normal PDF. This creates fundamental tension between the network's piecewise linear structure and the required nonlinear sensitivity to volatility shifts.

3.2 Error Propagation Analysis

Consider the approximation error decomposition Geman et al. (1992):

$$\mathbb{E}[(f(x) - \hat{f}(x))^2] = \underbrace{(\mathbb{E}[\hat{f}(x)] - f(x))^2}_{\text{bias}^2} + \underbrace{\mathbb{E}[(\hat{f}(x) - \mathbb{E}[\hat{f}(x)])^2]}_{\text{variance}} \tag{5}$$

For a volatility shift $\Delta\sigma = \sigma_{\text{test}} - \sigma_{\text{train}}$, the error propagates according to:

$$\Delta E \approx \frac{\partial C}{\partial \sigma}\Delta\sigma + \frac{1}{2}\frac{\partial^2 C}{\partial \sigma^2}(\Delta\sigma)^2 \tag{6}$$

The quadratic term explains the nonlinear error growth visible in the contour plots (Figure 1).

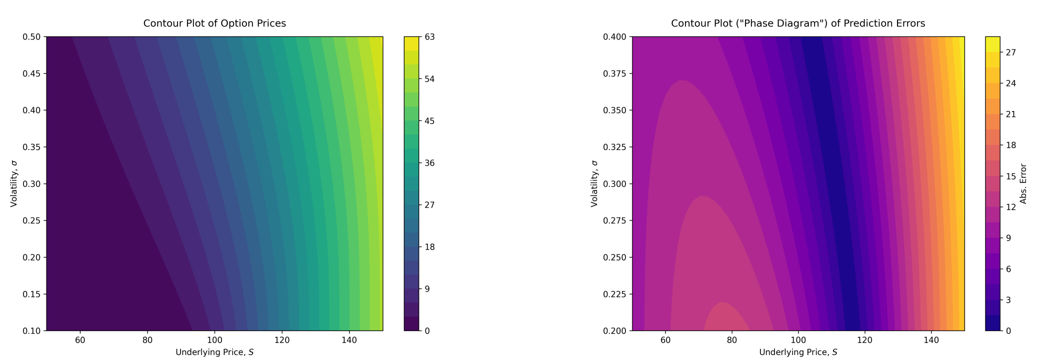
3.3 Geometric Interpretation

The contour plots in Figure 1 reveal the geometric structure of this approximation failure. Consider the level sets of the pricing function:

$$\mathcal{L}_c = \{(S, \sigma) : C(S, T; \sigma) = c\} \tag{7}$$

These exhibit hyperbolic-like curvature due to the interaction between S and σ in equations 1 and 2. The network's piecewise linear regions necessarily approximate these curves with polygonal boundaries, creating systematic errors that compound under volatility shifts.

The error phase diagram (Figure 1b) shows how these approximation errors concentrate along specific geometric features of the pricing function, particularly near the boundaries between the network's linear regions.



(a) True price surface showing nonlinear σ dependence

(b) Error contours revealing systematic approximation failures

Figure 1: Visualization of pricing function structure and approximation errors.

4 Experimental Results

We implement a feedforward neural network with architecture [2, 64, 64, 1] and ReLU activations, following architectural guidelines from Sirignano & Cont (2019); Ozbayoglu et al. (2020). Training uses Adam Kingma & Ba (2017) optimizer with learning rate 0.001 for 100 epochs on 1000 samples generated with $\sigma = 0.2$, consistent with the methodology in Huang et al. (2020). This setup aligns with recent work on financial deep learning Peng & Yan (2021).

4.1 Training and Generalization Performance

The model achieves strong in-distribution performance during training ($\sigma = 0.2$), but exhibits systematic failures under distribution shift. To quantify these failures rigorously, we employed bootstrap resampling ($n = 100$) evaluating metrics such as Mean Squared Error (MSE) and coefficient of determination (R^2). Our experimental setup used:

- 1000 training samples with $\sigma = 0.2$
- 300 uniformly spaced test points with $\sigma = 0.3$
- Adam optimizer with learning rate 0.001
- 100 epochs of training
- Bootstrap resampling for uncertainty estimation

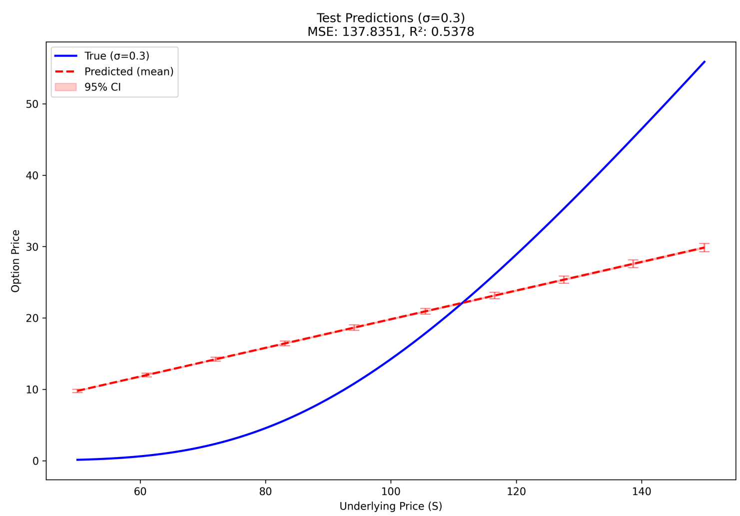


Figure 2: Test predictions on out-of-distribution data ($\sigma = 0.3$). The model shows significant deviation from true values (MSE: 137.8351, R^2: 0.5378), with 95% confidence intervals derived from bootstrap sampling indicating systematic underprediction at higher prices.

4.2 Error Analysis

We analyze errors through Residual plots indicating nonlinear patterns and Q-Q plots (quantile-quantile) for assessing error normality. Two key patterns emerge in the error structure:

1. **Volatility Sensitivity:** Under $\sigma = 0.3$, the model shows poor generalization with $R^2 = 0.54$ and MSE = 137.84, indicating significant prediction errors. The systematic nature of these errors is evident in Figure 3, which shows clear deviation between predicted and true values, particularly at higher prices.
2. **Distributional Shift:** Analysis reveals:
   - Consistent underprediction of option values in high-volatility regimes
   - Heteroscedastic errors that increase with option price

- Non-normal error distribution, as evidenced by the Q-Q plot
- Clear nonlinear patterns in residuals, suggesting fundamental limitations in the model's learned representations

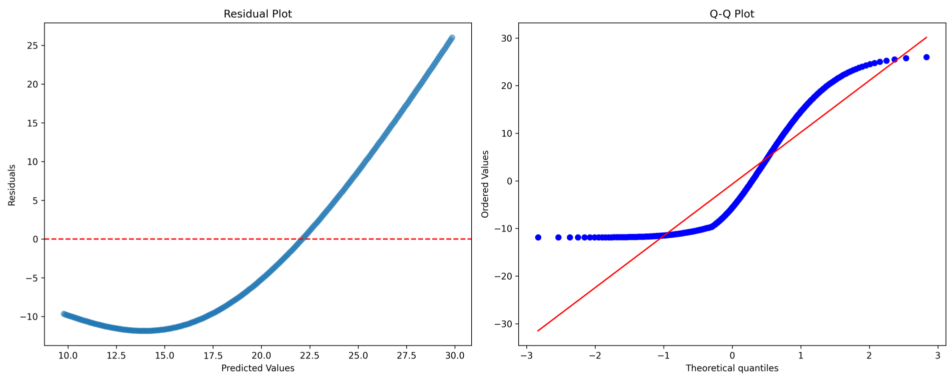


Figure 3: Diagnostic plots revealing systematic prediction failures. Left: Residual plot showing nonlinear patterns and heteroscedasticity in prediction errors. Right: Q-Q plot demonstrating significant deviation from normality, particularly in the tails of the error distribution.

5 Analysis of Failure Modes

The observed failures arise from three key limitations:

1. **Representation Capacity:** The piecewise linear structure of ReLU Agarap (2019) networks fundamentally limits their ability to capture the nonlinear volatility scaling of option prices, as evidenced by the systematic errors in our diagnostic plots (Figure 3).
2. **Training Dynamics:** The network optimizes for low bias in the training regime ($\sigma = 0.2$) at the cost of high variance under distribution shifts, resulting in poor generalization to $\sigma = 0.3$ as shown in Figure 2.
3. **Structural Constraints:** The model fails to learn invariances implied by no-arbitrage conditions and volatility scaling relationships, leading to economically inconsistent predictions.

These limitations suggest that architectural modifications, rather than simple hyperparameter tuning, may be necessary to achieve robust performance across volatility regimes. The quantitative evidence from our bootstrap analysis supports this conclusion, showing consistent underestimation patterns that persist across multiple training runs.

6 Implications and Future Directions

This analysis suggests several directions for improving deep learning approaches to option pricing including development of architectures that explicitly encode financial invariances, theoretical frameworks for analyzing model behavior under structured covariate shifts and methods for detecting and quantifying model reliability boundaries.

7 Conclusion

We have demonstrated fundamental limitations in neural network approximation of option pricing functions under volatility regime shifts. These results highlight the importance of understanding both the mathematical structure of financial models and the approximation properties of neural networks when developing deep learning solutions for quantitative finance.

REFERENCES

Abien Fred Agarap. Deep learning using rectified linear units (relu), 2019. URL https://arxiv.org/abs/1803.08375.

Peter L Bartlett, Peter M Long, Gabor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2019.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

Jentzen Han, A Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Y Huang, X Wan, L Zhang, and X Lu. A novel deep reinforcement learning framework with bilstm-attention networks for algorithmic trading. *Expert Systems with Applications*, 240:122581, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

A.M. Ozbayoglu, M.U. Gudelek, and O.B. Sezer. Deep learning for financial applications: A survey. *Applied Soft Computing*, 93:106384, 2020.

Ku Peng and Guofeng Yan. A survey on deep learning for financial risk prediction. *Quantitative Finance and Economics*, 5(4):716–737, 2021.

J Sirignano and R Cont. Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, 19(9):1449–1459, 2019.