Neuro-Hedge: Minimizing Hedging Error and Transaction Costs via Guided Reinforcement Learning

Technical Report

February 9, 2026

Abstract

This report details **Neuro-Hedge**, a Deep Reinforcement Learning (DRL) framework designed to optimize derivatives hedging strategies in markets with linear transaction costs. While traditional analytical models like Black-Scholes-Merton (BSM) assume frictionless trading, Neuro-Hedge employs a "Guided DDPG" architecture to balance risk mitigation against execution costs. By integrating domain knowledge through probabilistic Teacher Forcing and Reward Regularization, the agent achieves a superior mean P&L (-10.73) compared to the BSM baseline (-10.89) over 5,000 training episodes.

1 Introduction

Hedging financial derivatives aims to neutralize risk associated with underlying asset price movements. The standard BSM model prescribes continuous Delta hedging to eliminate risk, but this theoretical framework fails in real-world markets where transaction costs (spreads and commissions) erode portfolio value. Frequent rebalancing to minimize risk incurs high costs, creating a stochastic control problem that is analytically intractable. Neuro-Hedge addresses this by utilizing Deep Deterministic Policy Gradient (DDPG) to learn an optimal policy that balances hedging error and transaction costs. To overcome the "cold-start" problem common in financial RL, the system uses the analytical BSM solution as a prior to guide exploration.

2 Methodology

2.1 Market Environment Modeling

The market is simulated using Geometric Brownian Motion (GBM) for the underlying asset price S_t. The dynamics are governed by the stochastic differential equation:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{1}$$

where μ is the drift, σ is volatility, and W_t is a Wiener process. The target derivative is a European Call Option priced via the BSM formula $C(S, t)$.

2.2 MDP Formulation

The hedging problem is formulated as a finite-horizon Markov Decision Process (MDP) tuple (S, A, R, γ):

- **State Space (S):** A normalized vector $s_t = [\tau_t, \frac{S_t}{K}, \delta_{t-1}]$, where τ_t is time to maturity, S_t/K is moneyness, and δ_{t-1} is the agent's current position. Including the previous position δ_{t-1} is crucial for calculating the marginal cost of new actions.

- **Action Space (A):** The action $a_t \in [0, 1]$ represents the target hedge ratio (number of shares to hold). The actual trade executed is $\Delta a_t = a_t - \delta_{t-1}$.
- **Reward Function (R):** The composite reward function balances three objectives:

$$r_t = -|V_t - C_{BS}| - \lambda_{txn}|\Delta a_t|S_t c_{rate} - \lambda_{reg}|a_t - \Delta_{BS}| \tag{2}$$

  where the terms penalize hedging error (deviation from theoretical price), transaction costs, and deviation from the BSM Delta (Δ_{BS}), respectively.

2.3 Guided DDPG Architecture

The system employs an off-policy Actor-Critic method (DDPG).

- **Teacher Forcing:** To accelerate training stability, the agent selects the analytical BSM action with a probability P_{teach} that decays exponentially ($P_{teach}(e) = \max(0, \alpha^e)$). This ensures the replay buffer is populated with high-quality transitions early in training.
- **Network Structure:** The Actor uses hidden layers of [400, 300] units with ReLU activations and a Sigmoid output. The Critic processes state and action inputs through concatenated dense layers to estimate Q-values.

3 Experimental Results

3.1 Setup

The agent was trained for 5,000 episodes with the following parameters: Volatility ($\sigma = 20\%$), Risk-free rate ($r = 5\%$), and Transaction Cost Rate ($c_{rate} = 0.2\%$).

3.2 Performance Evaluation

The trained Neuro-Hedge agent was evaluated against a pure Black-Scholes baseline over 1,000 out-of-sample episodes.

- **Financial Performance:** The RL agent achieved a mean P&L of **-10.73**, outperforming the BSM baseline of **-10.89**. This indicates better efficiency in the presence of market friction.
- **Behavioral Analysis:** While the RL agent incurred slightly higher average transaction costs (≈ 0.50 vs 0.44), it achieved a better overall P&L. This suggests the agent learned to "spend" its transaction budget more wisely, trading only when the reduction in hedging error justified the cost.
- **Smoothing Effect:** Trajectory analysis shows the RL agent exhibits a "smoothing" behavior, avoiding small, costly adjustments that do not significantly reduce risk, effectively adapting to the cost structure.

4 Conclusion

Neuro-Hedge successfully demonstrates that a guided DRL agent can outperform traditional analytical models in realistic markets with transaction costs. By incorporating domain knowledge through Teacher Forcing and Reward Regularization, the agent converges to a robust policy that intelligently balances risk and cost.