Multi-Layer Perceptron for Optimal Equity Derivative Hedging

Panfeng Jiang, Zhekai Zhang, Junyu Zhang, Jose Daniel Cantu Cantu

March 1, 2025

Contents

Abstract

This report presents an in-depth discussion of our option trading strategy. Our approach encompasses rigorous data preprocessing, a novel labeling method based on a profit-and-loss (PnL) and exposure trade-off, the training of a multi-layer perceptron (MLP) network through extensive cross-validation, and a systematic daily decision-making process that meets targeted exposure levels.

1 Introduction

The world of algorithmic trading, particularly in the domain of options, requires robust data handling, precise modeling, and dynamic decision-making. Our strategy has been meticulously designed to address these challenges. In this report, we outline a multi-faceted approach that integrates:

- **Data Preprocessing:** A thorough cleaning and formatting routine that handles missing values, aligns time series data, and ensures continuity.
- **Labeling and Network Training:** A novel method for labeling options based on their relative performance, followed by the training of an MLP network with rigorous cross-validation to select optimal model parameters.
- **Daily Decision Logic:** A step-by-step algorithm that decides on the purchase and sale of options in order to meet a specified exposure target, thereby balancing risk and potential returns.

This report is structured to provide detailed insights into every stage of our methodology. We discuss the theoretical underpinnings of our approach, the practical challenges encountered, and the solutions implemented to ensure that the trading strategy is both effective and reliable.

2 Data Preprocessing

Data quality is the cornerstone of any successful trading strategy. In our approach, data preprocessing is divided into three distinct yet interrelated steps: data filling, data calculating, and data splitting.

2.1 Data Filling

Financial data, especially in options trading, is often plagued by gaps and inconsistencies. In our dataset, we observed that some options do not appear continuously over the expected date range. This irregularity can be attributed to various factors such as low trading volume, market anomalies, or reporting issues. To avoid erroneous predictions or skewed model training, it is essential to address these gaps in the data.

Our approach to data filling involves the following steps:

1. **Identifying Available Dates:** For each option, we extract the list of dates on which data is recorded. This step ensures we know exactly where the data is available.

2. **Filling Missing Values:** For any date where an option's data is missing, we insert a record with a price of 0 and a size of 0. This choice is deliberate: a zero value indicates that no transaction occurred, rather than interpolating values that might introduce bias.

An important nuance in our data filling strategy is the treatment of weekends. Since financial markets are closed during weekends, no trading occurs, and thus, we intentionally exclude these days from the data filling process. This ensures that our model is not misled by artificial zero entries on non-trading days.

2.2 Data Calculating

For each unique option, our goal is to calculate two key metrics over its life cycle—from the first appearance to its expiration:

- **Profit and Loss (PnL):** This metric reflects the net gain or cost associated with holding an option. The PnL is computed by comparing the option's price changes over time.

- **Exposure:** Exposure quantifies the sensitivity or risk associated with the option. It can be viewed as the amount of capital at risk in a particular option.

The calculation involves:

1. **Initialization:** For each option, identify the date of first appearance and the expiration date.

2. **Iterative Calculation:** Using the filled dataset, compute the daily PnL by taking the difference in prices from one day to the next. Similarly, calculate the exposure based on the option size and other relevant risk factors.

3 Labeling and Network Training

In this section, we elaborate on the process of labeling our dataset and training a neural network model to predict optimal trading decisions.

3.1 Labeling

The primary goal of labeling is to quantify the desirability of each option by combining its risk (exposure) and potential return (PnL). A lower label value indicates a more preferred option, serving as a target for our predictive model.

We adopt a greedy strategy based on the assumption that all purchased options can be liquidated the following day. This simplifying assumption allows us to focus on immediate profitability and risk management, transforming the complex evaluation of option portfolios into a manageable day-to-day decision process.

The central metric used for labeling is the ratio of exposure to PnL on a specific date. The rationale behind this metric is twofold:

1. **Positive PnL and Exposure:** For options with positive values, a smaller ratio (i.e., lower exposure relative to profit) is preferred.

2. **Negative PnL and Exposure:** In cases where the values are negative, a larger absolute ratio is preferable, as it indicates a lower relative loss compared to the risk taken.

Subsequently, a mapping function is applied to convert this ratio into a scalar label. The mapping is designed such that:

- Smaller labels denote more desirable options.
- The mapping function differentiates between positive, negative, and zero ratios in a way that preserves the intended order of preference.

While the precise mathematical formulation of the mapping function is embedded in our code, the conceptual idea is as follows:

- For **positive ratios**, the function is monotonically increasing in absolute value, but options with smaller ratios are rewarded with lower labels.
- For **negative ratios**, the function inverts the scale such that higher absolute values result in a lower label, reflecting the relatively lower risk per unit of loss.
- A default label is assigned to options with a ratio of zero, which are considered the least preferred.

$$f(\text{ratio}) = \begin{cases} 1 - \dfrac{1}{1+\text{ratio}}, & \text{if ratio} > 0, \\ 1 + \dfrac{1}{1-\text{ratio}}, & \text{if ratio} < 0, \\ 3, & \text{otherwise.} \end{cases}$$

This mapping ensures that the final labels effectively capture the trade-off between risk and return, enabling the MLP to learn a meaningful ordering of options.

3.2 Network Training

The predictive model is based on a multi-layer perceptron (MLP). The architecture is chosen for its flexibility and ability to capture non-linear relationships within the data.

The training process involves:

1. **Data Feeding:** The preprocessed and labeled data is fed into the network.

2. **Loss Function:** We employ a Mean Squared Error that penalizes deviations between the predicted labels and the true labels.

3. **Optimization:** Adam is used to update network weights iteratively.

4. **5-Fold Cross-Validation:** To ensure model robustness, we implement a 5-fold cross-validation scheme. This involves splitting the data into five subsets, training on four subsets while validating on the remaining one, and rotating the validation subset. The model with the best average performance is selected as the final model.

Using an MLP for this task provides several advantages:

- The non-linear modeling capacity of the MLP allows for capturing intricate relationships between the input features and the label.

- Cross-validation ensures that the model is not overfitting to the training data and can generalize well to unseen market conditions.

- The approach is scalable and can be extended with additional features or more complex architectures if needed.

4 Daily Decision Logic

The daily decision-making process is designed to dynamically adjust our holdings to meet a targeted exposure level. This section details the step-by-step logic that drives our trading strategy.

Our primary aim in daily decision-making is to select and execute trades that align our portfolio with a pre-specified exposure target. This involves both buying new options and managing existing sell orders to achieve the desired risk profile.

4.1 Step 1: Managing Existing Sell Orders

4.1.1 Sell Order Dictionary

We maintain a comprehensive dictionary of all options that are currently available for sale. This dictionary is continuously updated based on the evolving market conditions and the trading history of each option.

- **Purpose:** The dictionary serves as a quick reference to assess whether current holdings are sufficient to meet the exposure target.

- **Operation:** Before any new trades are executed, the algorithm checks if the cumulative exposure from the sellable options meets or exceeds the target. If so, these options are sold, and the portfolio's exposure is adjusted accordingly.

4.1.2 Decision Flow

The logic can be summarized as:

1. **Evaluate Holdings:** Calculate the total exposure available from the current holdings.

2. **Compare with Target:** If the current exposure meets the target, initiate the sale of the corresponding options.

3. **Update Holdings:** After executing sell orders, update the holdings to reflect the new exposure status.

4.2 Step 2: Iterative Purchase and Order Adjustment

4.2.1 Label-Based Option Selection

Options are iterated over in ascending order of their labels (i.e., from the most preferred to the least preferred). This ordering ensures that the algorithm prioritizes options with the best risk-return trade-off.

- **Withdrawal of Sell Orders:** If an option currently exists in the selling dictionary, the planned sell orders for that option are first withdrawn. This allows for reallocation of the option in a manner that better fits the current exposure target.

- **Direct Purchase:** For options not present in the sell dictionary, a direct purchase is executed.

4.2.2 Exposure Calculation and Trade Execution

For each option, the algorithm performs the following computations:

1. **Exposure Adjustment:** Determine how much buying (or withdrawing sell orders) will contribute towards reaching the exposure target.

2. **Conditional Execution:**

   - If the calculated action exactly meets the target, execute the trade for the required amount.
   - If the target is not met, execute the maximum allowable trade for that option and then proceed to the next one.

3. **Iterative Process:** The process continues until the cumulative exposure matches the desired target. This iterative decision-making process ensures that the portfolio is dynamically balanced based on real-time data.

4.2.3 Daily Update and Iteration

At the end of each trading day, the algorithm:

- Updates the sell dictionary to reflect any changes in option availability.
- Recalculates cumulative exposure based on newly executed trades.
- Prepares the system for the next day's trading decisions by resetting temporary variables and logging performance metrics.

This cycle ensures that the trading strategy is continuously aligned with market conditions and exposure targets.

5 Conclusion

In this comprehensive report, we have detailed an end-to-end option trading strategy that spans data preprocessing, innovative labeling based on a PnL-exposure trade-off, MLP network training with rigorous cross-validation, and a dynamic daily decision-making process. Our methodology is designed to optimize both profitability and risk management, ensuring that trading decisions are made based on sound quantitative principles.

The multi-layer perceptron model is a key component in our strategy, enabling us to predict the most favorable options based on historical data and computed metrics. Meanwhile, the daily decision logic provides a systematic approach to managing both buy and sell orders, ensuring that our exposure targets are met in a controlled and efficient manner.

Future work could involve exploring more complex network architectures, integrating additional market indicators, and refining the labeling mechanism further. By continuously iterating on these components, we aim to enhance the performance and robustness of our trading strategy in increasingly volatile markets.