Deep Learning for Portfolio Optimization Using Transformer-Based Financial Time-Series Models

Submitted By: Dibakar Bhowal

Submitted To: Dr. Chaojiang Wu

Final Project

Advanced Machine Learning

BA-64061-001

Abstract

Deep Learning has become a game-changer in financial forecasting and portfolio building - giving us the ability to model non-linear relationships, temporal structures and cross-asset interactions that traditional quantitative methods often fails to capture. This study looks into the very latest Deep Learning techniques - specifically the Transformer architecture and long short-term memory (LSTM) networks, for weekly portfolio weight optimization across exchange-traded funds (SPY, QQQ, IWM, RSP) and individual equities (MSFT, AAPL, TSLA, NVDA). Using a feature-rich data pipeline including daily and weekly returns, moving averages,and rolling volatility, we compare each model's ability to maximize risk-adjusted performance. Results show that although both architectures can learn meaningful temporal patterns, the Transformer outperforms the LSTM in Sharpe ratio, stability, and generalization. This paper concludes with an industry analysis of deep learning in finance and discusses future research directions, including model interpretability, alternative objective functions and reinforcement-learning-based allocation strategies.

1. Introduction

Deep learning has rapidly evolved into a powerful methodology for analyzing high-dimensional and nonlinear datasets across domains such as computer vision, speech, natural language processing and financial forecasting (LeCun et al., 2015; Goodfellow et al., 2016). Financial markets present a unique modeling challenge due to noise, non-stationarity, volatility clustering and structural regime changes (Tsay, 2010). Traditional econometric and quantitative models often fail to capture these complexities, motivating the use of modern neural architectures such as recurrent networks, convolutional networks and attention-based Transformers. Portfolio optimization, the allocation of asset weights that maximize expected return for a given level of risk, represents a critical application of deep learning in finance. This project focuses on using deep learning to directly output portfolio weights each week, trained using a loss function that maximizes the Sharpe ratio.

2. Literature Review: Deep Learning for Financial Time-Series

2.1 Recurrent Neural Networks and LSTMs: LSTMs were introduced to solve long-term dependency issues in vanilla RNNs (Hochreiter & Schmidhuber, 1997). They have been widely adopted for stock prediction, volatility estimation, and macroeconomic forecasting. However, LSTMs process data sequentially and struggle with long-range dependencies present in financial data.

2.2 Convolutional Architectures: CNNs have been applied to time series by interpreting input sequences as structured matrices (Zhang et al., 2017). They capture local temporal patterns but struggle to represent long-term cross-asset interactions.

2.3 Attention Mechanisms and Transformers: The Transformer architecture (Vaswani et al., 2017) revolutionized NLP by enabling direct modeling of long-range relationships without recurrence. Recent literature applies Transformers to financial forecasting due to: Multi-asset interactions (cross-sectional attention) Ability to capture long-term dependencies Parallel sequence processing Superior empirical performance (Zhang et al., 2022; Kim et al., 2021)

2.4 Deep Learning for Portfolio Optimization: Recent work proposes end-to-end models that produce portfolio weights via softmax outputs, trained on utility or Sharpe-ratio objectives (Moody & Saffell, 2001; Jiang et al., 2017). The present project follows this paradigm.

3. Methodology

3.1 Data Sources

Historical data was collected using *yfinance*, consisting of:

- ETFs: **SPY, QQQ, IWM, RSP**
- Stocks: **AAPL, MSFT, TSLA, NVDA**

Daily prices were converted into:

- Daily simple returns
- Weekly log returns
- 5-day moving average
- 20-day moving average
- 20-day rolling volatility

These features were merged into a unified dataset and aligned by date.

3.2 Preprocessing Pipeline

3.2.1 Scaling

Two *StandardScaler* instances were fit only on the training set to prevent data leakage:

- Feature scaler
- Target (weekly returns) scaler

This follows best practice for time-series modeling (Hyndman & Athanasopoulos, 2018).

3.3 Sequence Construction

A custom function created fixed-length input sequences:

- Sequence length = 16 weeks
- Each sequence produced the next-week returns as targets.

This transforms the dataset into a supervised learning structure suitable for LSTM and Transformer models.

3.4 Model Architectures

3.4.1 Transform-Based Portfolio Model

Key characteristics:

- Input: `(batch, sequence_length, feature_dim)`
- TransformerEncoderLayer:
  - `d_model = 64`
  - `nhead = 4`
  - `num_layers = 2–4` depending on hyperparameter search
- Output: A linear head producing asset weights
- Softmax activation ensures weights sum to 1

The self-attention mechanism allows the model to learn cross-asset dependencies and long-range temporal dynamics.

3.4.2 LSTM Portfolio Model

- Two LSTM layers with hidden size 64
- Final linear layer -- softmax
- Captures local temporal dependencies but lacks expansive context modeling

3.5 Loss Function

A differentiable approximation of the **Sharpe Ratio** was used:

Loss = $- \mu / (\sigma + 1e\text{-}6)$

Where:

- μ = mean portfolio return
- σ = standard deviation of returns

This encourages high return and low volatility.

3.6 Hyperparameter Tuning

A manual grid search optimized:

- Hidden size

- Number of layers
- Learning rate
- Dropout
- Optimizer (Adam)

Each combination was trained with early stopping based on validation Sharpe ratio.

4. Results and Analysis

4.1. Hyperparameter Tuning and Final Configurations

A reduced grid search was executed to find the optimal settings for both the Transformer and LSTM models, aiming to maximize the validation Sharpe Ratio and mitigate overfitting via early stopping. Both models were trained for a maximum of 50 epochs.

The best performing Transformer model was achieved with a learning rate (lr) of 0.0001, a weight decay of 1e−05, and a patience of 10 for early stopping. This configuration ultimately delivered a cumulative return (CR) of 139.42% on the test set, corresponding to a Sharpe Ratio (SR) of 0.2723 with a volatility (σ) of 3.36%.

The best performing LSTM model used a learning rate (lr) of 0.001, a weight decay of 0.0001, and a patience of 7. This model achieved a test cumulative return of 112.26%, corresponding to a Sharpe Ratio of 0.2641 with a lower volatility of 2.97%.

4.2. Comparative Performance (Transformer vs. LSTM)

The Transformer model demonstrated superior out-of-sample generalization and overall performance when evaluated on the held-out test set, primarily due to its ability to maximize mean portfolio return.

The Transformer significantly outperformed the LSTM in total cumulative return (CR: 139.42% vs. 112.26%) and yielded a higher risk-adjusted return (Sharpe Ratio: 0.2723 vs. 0.2641). While the LSTM model exhibited slightly lower portfolio volatility (2.97% vs. 3.36% for the Transformer), the Transformer's higher mean return was sufficient to confirm its better overall performance.

This outcome suggests that the self-attention mechanism within the Transformer architecture is more effective at modeling the complex, nonlinear cross-asset dependencies and long-range temporal structures characteristic of financial data than the LSTM's sequential processing approach. The Transformer also showed better stability during training and lower loss volatility across hyperparameter combinations, while the LSTM struggled more with volatility and oversensitivity to hyperparameters.

4.3. Risk and Drawdown Analysis

The analysis of portfolio drawdowns is critical for evaluating risk management. The cumulative returns plot visually confirms the Transformer's more consistent growth, particularly in periods like mid-2024, where its returns sustained an upward trajectory while the LSTM's performance was flatter.

Analyzing the Portfolio Drawdowns reveals that both models experienced similar maximum drawdown magnitudes (peaking around −13% to −14% in late 2023). Crucially, the Transformer's drawdown path exhibited fewer and shallower drawdowns in less extreme periods, indicating a potential advantage in volatility control and risk management during typical market fluctuations.

4.4. Feature Importance Diagnostics

A Random Forest Regressor was used to assess the aggregated importance of the engineered features (moving averages and volatility) for predicting the next-week returns across all eight assets.

The analysis clearly highlighted the dominant role of asset-specific volatility features as the top aggregated predictors, reinforcing the importance of risk factors as direct inputs for a Sharpe-ratio-optimized model. The three most impactful features aggregated across all assets were:

1. TSLA_volatility: 0.797298
2. AAPL_volatility: 0.652182
3. NVDA_volatility: 0.631022

In addition to volatility, momentum signals were also prominent. Notable features derived from moving averages included TSLA_long_ma (0.371963) and TSLA_short_ma (0.300706). This outcome confirms that the deep learning model’s decisions rely heavily on both momentum/trend (moving averages) and risk signals (volatility) to generate superior risk-adjusted returns.

The Transformer model achieved superior Sharpe ratio, stability, and generalization compared to the LSTM model. The self-attention mechanism enabled the model to capture nonlinear cross-asset dependencies and long-range temporal structures more effectively. LSTM performance was weaker due to vanishing gradients and difficulties modeling regime changes. Random forest-based feature importance confirmed that moving averages and volatility were significant predictors. The Transformer model's cumulative returns were smoother with better volatility control.

5. Industry Applications of Deep Learning

Deep learning is widely applied in multiple industries: finance (algorithmic trading, risk management, fraud detection), healthcare (medical imaging, diagnostics), transportation (autonomous vehicles, route optimization), and security (threat detection, biometrics). The same architectures explored in this project, LSTMs and Transformers, are used extensively across these applications.

6. Future Directions

Future research may incorporate Temporal Fusion Transformers, reinforcement learning approaches, higher-frequency data, alternative data sources, or interpretability tools such as SHAP values. Real-world deployment additionally requires modeling transaction costs, slippage, and liquidity constraints.

7. Conclusion

This research demonstrates that Transformers provide a powerful deep learning architecture for portfolio optimization, outperforming LSTMs in Sharpe ratio, cumulative return and stability. By leveraging attention mechanisms and sequence modeling techniques, Transformer-based models can capture complex financial dependencies and support more robust portfolio allocation strategies.

References

- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice*. OTexts.

- Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*.

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444.

- Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4), 875–889.

- Tsay, R. S. (2010). *Analysis of financial time series*. Wiley.

- Vaswani, A., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

- Zhang, Y., Aggarwal, C., & Qi, G. J. (2017). Stock price prediction via CNNs. *IEEE ICDM*.