Physics-Informed Neural Networks for Portfolio Optimization

Table of Contents

---

1. Introduction

This project focuses on solving a portfolio optimization problem using both deep learning and classical time series techniques. We aim to model the optimal investment strategy by solving the Hamilton-Jacobi-Bellman (HJB) equation using Physics-Informed Neural Networks (PINNs), forecast asset returns using Long Short-Term Memory (LSTM) networks, and compare the strategy against benchmarks. ARIMA and GARCH are included to provide traditional econometric analysis.

---

2. Problem Statement

The goal is to optimize investment decisions over time by maximizing terminal utility. The investment problem can be framed using the HJB PDE derived from dynamic programming. We train PINNs to learn the value function and policy, and integrate LSTM forecasts for dynamic strategy simulation. Performance is compared with FDM, fixed strategies, and real market benchmarks.

---

3. Theoretical Background

- The Merton problem provides a theoretical basis for continuous-time portfolio optimization under CRRA utility.

- The HJB PDE describes the value function evolution under optimal policy.
- PINNs embed PDE constraints into the training objective.
- LSTMs capture sequential patterns for forecasting.
- ARIMA models autocorrelated returns; GARCH captures volatility clustering.

---

4. Dataset and Preprocessing

- Data: MSFT, AAPL, GOOG, AMZN, and SPY from Yahoo Finance (2020–2025)
- Returns: Daily log returns computed after forward-filling missing data
- Stationarity: ADF test applied prior to ARIMA/GARCH fitting
- Prepared Inputs: Sequences for LSTM, return vectors for time series modeling

---

5. Methods Employed

5.1 Physics-Informed Neural Networks (PINNs)

Solve the HJB PDE by enforcing its residual and terminal condition via a loss function. The model approximates the value function V(t,w).

5.2 Finite Difference Method (FDM)

Discretize the HJB using backward Euler scheme on a time-wealth grid and solve it recursively.

5.3 LSTM-Guided Dynamic Allocation

Predict next-day returns using a rolling sequence of previous daily returns. These predictions help rebalance the portfolio.

5.4 ARIMA and GARCH Modeling

Fit ARIMA(1,0,1) to model return autocorrelation and GARCH(1,1) to estimate return volatility over time.

---

6. Model Architectures

6.1 PINN Architecture

- Input: time `t`, wealth `w`
- Layers: 3 hidden layers, 50 neurons each, tanh activation
- Output: value function `V(t, w)`
- Loss: PDE residual + terminal utility condition
- Differentiation: via JAX automatic differentiation

6.2 LSTM Architecture

- Input: 20-day return sequence
- Layer: 1 LSTM layer with 64 units
- Output: 1-day ahead return forecast
- Activation: ReLU for output

6.3 FDM Solver

- Time and wealth discretization
- Backward Euler in time, central difference in wealth
- Iterative update from terminal to initial time

6.4 ARIMA and GARCH

- ARIMA(1,0,1): fitted using AIC
- GARCH(1,1): models time-varying volatility
- Both used to support portfolio decisions and evaluate risk

---

7. Model Training and Implementation Details

- PINN: Trained for 5000 epochs using Adam optimizer
- LSTM: Batch size 32, dropout 0.2, 64 hidden units
- Time Series Models: Maximum Likelihood Estimation used for ARIMA/GARCH
- Libraries: JAX for PINN; TensorFlow for LSTM; statsmodels for ARIMA/GARCH

---

8. Model Evaluation and Visualization

8.1 Training Diagnostics

Loss curves for PINN show convergence after about 1000 epochs. HJB residual and terminal loss reduced steadily.

8.2 Learned Value Function vs Exact Solution

PINN value heatmap aligns well with analytical solution. Absolute error lowest at mid-wealth, higher at extremes.

8.3 Policy Visualization

Optimal policy $\pi^*(t, w)$ decreases with wealth and time, as expected. High risk-taking for low wealth, early time.

8.4 Portfolio Simulation

PINN + LSTM model outperformed fixed policy and SPY benchmark by end of period (2024).

8.5 Drawdown and Risk Metrics

Maximum drawdown in 2022 was 40%, but recovery was strong. Final Sharpe ratio higher than SPY.

---

9. Time Series Analysis

- Stationarity confirmed with ADF test
- GOOG showed best ARIMA performance (lowest AIC)
- MSFT showed highest GARCH stability (lowest volatility persistence)
- AAPL had the most volatility

---

10. Comparative Analysis

PINN vs FDM

Aspect	PINN	FDM
Smoothness	High	Grid-based
Boundary Handling	Soft	Hard
Accuracy	Moderate	High
Computational Cost	High (training)	Medium

Portfolio Strategies

Strategy	Final Wealth	Sharpe Ratio	Max Drawdown	Volatility
PINN+LSTM	$220,000	0.90	40%	30%
SPY	$160,000	0.65	30%	18%
Fixed 60%	$180,000	0.75	35%	22%

ARIMA and GARCH

Stock	AIC	α	β	α+β	Notes
GOOG	1200	0.10	0.85	0.95	Moderate risk
MSFT	1150	0.08	0.88	0.96	Most stable
AAPL	1100	0.09	0.87	0.96	Most volatile

---

11. Challenges Faced

PINN Training and Loss Balancing:

Training the PINN was challenging due to the need to weight the HJB residual loss against the terminal condition. An imbalance would skew learning toward one component, so we manually tuned weights and experimented with network depth. The model was also slow to converge and required GPU acceleration. Balancing these losses properly was important for the PDE solution to be learned correctly, and many epochs were needed for stability.

LSTM Sequence Modeling and Overfitting:

The LSTM model risked overfitting due to the limited amount of historical stock data. We addressed this by applying dropout, early stopping, and choosing shorter sequences. Despite these steps, capturing real market structure accurately was still difficult. The model sometimes captured noise rather than meaningful return signals.

FDM Stability and Grid Sensitivity:

The FDM solver's stability depended heavily on the choice of grid. Coarse grids led to inaccurate approximations near boundaries and introduced numerical oscillations. Fine grids improved accuracy but increased computation time. Implementing boundary conditions also required careful tuning and monitoring to prevent instability.

Data Preprocessing Challenges:

Real stock market data includes missing days, outliers, and changes in volatility over time. We had to preprocess this data carefully to remove inconsistencies, forward-fill missing entries, and ensure stationarity for time series models. Aligning return series across assets was crucial for LSTM input preparation and for portfolio simulations to be valid.

Integration of Model Outputs into Simulation:

One of the main difficulties was using model outputs (from PINN or LSTM) in a realistic simulation framework. Since forecasts were generated at time t and used at time t+1, there was a lag in action that reduced performance. Moreover, we did not model transaction costs, slippage, or liquidity effects, which are all present in real markets. These limitations may have overestimated portfolio performance. Future implementations should incorporate such realistic frictions.

---

12. Recommendations and Future Directions

12.1 Integration with Broader Financial Systems

- Deploy models in robo-advisors or quant strategies
- Connect PINN policies to automated trading engines

- Stream LSTM predictions into live dashboards

12.2 Real-World Applications

- Add trading fees and risk controls to simulation
- Use live APIs to automate investment based on model signals
- Apply in fintech tools for retail or institutional investors

12.3 Model Improvements

- Try attention-based LSTM or Transformer networks
- Use adaptive loss balancing for better PINN convergence
- Explore GARCH-PINN hybrid models for better volatility modeling
- Perform rigorous hyperparameter search and model ensembling

---

13. Glossary of complex Terms

Portfolio: A group of investments held together by an investor.
Risky Asset / Risk-Free Asset: A risky asset (like a stock) can go up or down a lot. A risk-free asset (like a bond) gives a fixed, safe return.
Utility Function: A math function representing investor happiness. We want to maximize it.
HJB Equation: A math formula used to find the best way to invest over time.
Value Function: Represents how happy an investor will be depending on their wealth and time left.
Optimal Policy (π)*: The best investment choice at a given moment.
Drawdown: The drop from the highest point of portfolio value to the lowest.
Sharpe Ratio: Return compared to risk. Higher is better.
Volatility: How much asset prices move around.
Log Returns: A way of measuring changes in prices that's more accurate than percent changes.
ARIMA: A model that predicts future returns using past trends and errors.
GARCH: A model that estimates future risk or volatility.
Benchmark: Something we compare our strategy to (like SPY).
Rebalancing: Adjusting the portfolio to keep the same investment proportions.

---

14. Conclusion

This project demonstrated that PINNs and LSTM can be combined to solve dynamic portfolio problems effectively. The PINN solves the HJB accurately, and LSTM enhances real-time decision-making. The model outperforms fixed strategies and SPY, while classical methods (ARIMA, GARCH) validate its behavior. The approach offers both theoretical and practical benefits for future financial modeling.

---

15. References

- *Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987–1007.*

- *Merton, R. C. (1971). Optimum consumption and portfolio rules in a continuous-time model. Journal of Economic Theory, 3(4), 373–413.*

- *Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707.*

- *Tsay, R. S. (2010). Analysis of financial time series (3rd ed.). Wiley.*

- *Plain English. (2023, February 24). Graph neural networks (GNNs) + physics-informed neural networks (PINNs) for stock market. Medium.*

- *Kundu, A., Bhardwaj, A., Das, D., & Chakraborty, A. (2022). Large-scale time-varying portfolio optimization using graph attention networks. Proceedings of the AAAI Conference on Artificial Intelligence, 36(1), 784–792.*