Deep Learning for Portfolio Optimization

Rudrajit Dey
rudrajit2906@gmail.com

Abstract

We adopt deep learning models to directly optimise the portfolio Sharpe ratio. The framework we present circumvents the requirements for forecasting expected returns and allows us to directly optimise portfolio weights by updating model parameters. Instead of selecting individual assets, we trade Exchange-Traded Funds(ETFs) of market indices to form a portfolio. Indices of different asset classes show robust correlations and trading them substantially reduces the spectrum of available assets to choose from. We compare our model with Naive and Markowitz theoretical models to show how it performs better.

Contents

1 Introduction

Portfolio Optimization is an essential component of trading systems. The optimisation aims to select the best asset distribution within a portfolio in order to maximise returns at a given risk level. This theory was pioneered by Markowitz in his work [1] and is the foundation of Modern Portfolio Theory. The main benefit of constructing such a portfolio comes from the promotion of diversification that smoothes out the equity curve, leading to a higher return per risk than trading an individual asset.

Despite the undeniable power of such diversification, it is not straightforward to select the "right" asset allocations in a portfolio, as the dynamics of financial markets change significantly over time. This is where deep learning comes in. Since it is not a deterministic model, it learns to adapt to the market changes and maximize our returns.

2 Literature review

The primary paper that I wanted to implement was [3]. In this paper they do not calculate expected returns and directly bypass this forecasting step to directly obtain asset allocations. This is where I found the idea of directly optimizing the Sharpe ratio thus maximizing return per unit risk. I also had an overlook into [2] so as to implement my original idea of optimizing portfolio along with proportional transaction cost. However, this was beyond my bounds of knowledge. I collected information about LSTM models from the internet and class notes.

3 Proposed methodology

We start with the objective function which lies at the heart of this problem.

The Sharpe ratio is used to gauge the return per risk of a portfolio and is defined as expected return over volatility (excluding risk-free rate for simplicity):

$$L = \frac{E(R_p)}{\mathrm{Std}(R_p)} \tag{1}$$

where $E(R_p)$ and $\mathrm{Std}(R_p)$ are the estimates of the mean and standard deviation of portfolio returns. We maximize this ratio or equivalently minimize the negative of the ratio.

4 Experimental result

4.1 Dataset

We use **adjusted close** for four market indices: **US total stock index (VTI), US aggregate bond index (AGG), US commodity index (DBC)** and **Volatility Index**

(VIX). These are popular Exchange-Traded Funds (ETFs) that have existed for more than 15 years. trading indices offers advantages over trading individual assets because these indices are generally uncorrelated resulting in diversification (see figure). A diversified portfolio delivers a higher return per risk and the idea of our strategy is to have a system that delivers good reward-to-risk ratio. Our dataset ranges from **30/04/2010** to **30/04/2025** and contains daily observations.

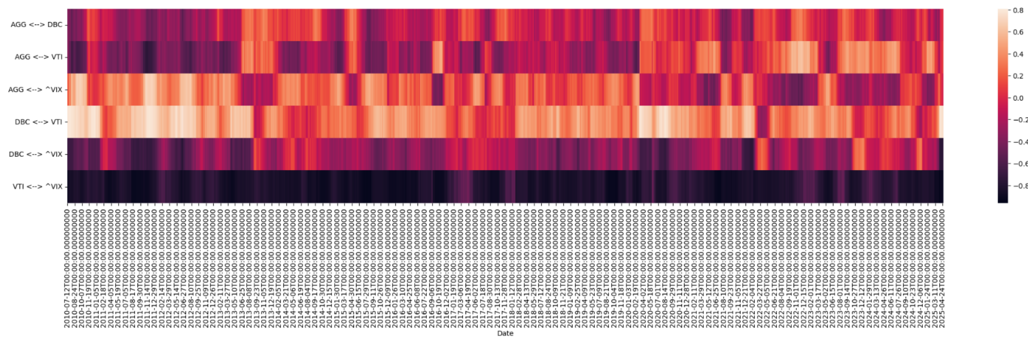

Figure 1: Heatmap for rolling correlations between different index pair.

We calculate the returns for each item and find out whether they are normally distributed, since the Sharpe Ratio assumes normal distribution of returns. Since, it is not we proceed to use the StandardScaler() to standardize training, validation and testing data which is split in the following manner:

- Training: 30-04-2010 to 01-05-2020 (2519, 4).
- Validation: 02-05-2020 to 02-02-2022 (443,4).
- Testing: 03-02-2022 to 30-04-2025 (811,4).

We then proceed to build sequences for LSTM with a lookback window of 50 days as mentioned in the paper.

4.2 Model Architecure

The model we used is as follows:

Layer (type)	Output Shape	Param #
Input (InputLayer)	(None, 50, 8)	0
lstm (LSTM)	(None, 64)	18,688
dropout (Dropout)	(None, 64)	0
dense (Dense)	(None, 4)	260

Total params: 18,948 (74.02 KB)

Trainable params: 18,948 (74.02 KB)

Non-trainable params: 0 (0.00 B)

Figure 2: LSTM Model, direct optimization.

- **Input Layer:** We have 4 assets in our portfolio. To form the input we concatenate the close prices and daily returns with a lookback of 50 and mini-batch size. Thus input shape is: (64, 50, 8).
- **Hidden Layers:** The LSTM operates with a cell structure that has gate mechanisms to summarise and filter information from its long history, so the model ends up with fewer trainable parameters and achieves better generalisation results.
- **Output Layer:** In order to construct a long-only portfolio, we use the softmax activation function for the output layer, which naturally imposes constraints to keep portfolio weights positive and summing to one. The number of output nodes 4, is equal to the number of assets in our portfolio, and we can multiply these portfolio weights with associated assets' returns to calculate realised portfolio returns (R_p). Once realised returns are obtained, we can derive the Sharpe ratio and calculate the gradients of the Sharpe ratio with respect to the model parameters and use gradient ascent to update the parameters.

4.3 Training Scheme

In this work, we use a single layer of LSTM connectivity, with 64 units, to model the portfolio weights and thence to optimize the Sharpe ratio. The Adam Optimizer is used for training our network, and the mini-batch size is 64. We train the model for 20 epochs and record the training loss and validation metric after each epoch.

4.4 Experiment Results

We provide the Sharpe Ratio comparison on the test set:

1. **Equal Weights:** 0.5809
2. **Markowitz Weights:** 1.87
3. **LSTM Weights:** 1.273

We also provide the cumulative returns for all three for comparison:

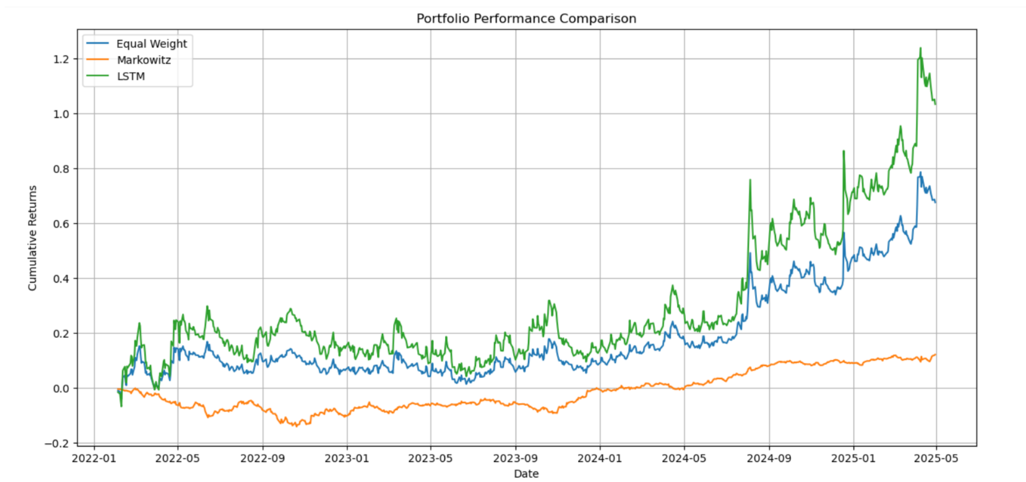

Figure 3: Portfolio Cumulative Returns.

5 Summary

We see that our LSTM model performed better than the Markowitz and the Naive model in terms of Expected returns, which means on average it will make us more money, however, it had a much higher variance which resulted in a smaller Sharpe Ratio. This can be explained by factors like:

- The LSTM model was trained from 2010 to start of 2020 which was mainly before the COVID crash while the testing was done on the recovery and also the current huge crash that the market saw due to the Tariff changes. This means the LSTM model learnt from a relatively much stronger market while it was tested on a highly volatile market. These might have resulted in a much higher Volatility and thus a higher variance in the returns. Whereas the Markowitz is somewhat oblivious to the volatility due to theoretical bounds and thus waivered better than the LSTM model.
- Lack of complexity in the model, with only one LSTM layer. Future work can include introducing Temporal Attention and even Deep-Reinforcement Learning.

- The paper [3] uses Volatility scaling as well as a transactional costs which scale their positions based on market-voatility and with No volatility scaling and minimal cost-rate, their Sharpe Ratio value was **1.858**.

References

[1] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[2] Weiwei Zhang and Chao Zhou. Deep learning algorithm to solve portfolio management with proportional transaction cost. In *2019 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*, pages 1–10, 2019.

[3] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep learning for portfolio optimization. *The Journal of Financial Data Science*, 2(4):8–20, August 2020.