

Portfolio Optimization with Deep Learning – Executive Summary

Team: Ahmet Özkan Demir, Amey Kaloti, Kwok Wai Ma

1. Overview

The goal of portfolio optimization is to select the best asset distribution among different financial products based on certain objectives. These objectives usually include maximizing the expected return, minimizing volatility, and/ or achieving a certain return rate in a fixed time window.

- Stakeholders: Investment firms, finance companies, investors
- KPI: Maximizing cumulative return and Sharpe ratio of the portfolio

2. Data Collection

A portfolio of the following 20 stocks is considered: (Google, Microsoft, Tesla, Citigroup, Costco, Walmart, Amazon, Nvidia, AMD, Meta Platforms, JP Morgan, Exxon Mobil, Marathon Oil, Johnson & Johnson, Procter & Gamble, Pfizer, Eli Lilly And Co, UnitedHealth Group, AT&T, McDonald's). The prices of these stocks are downloaded from Yahoo Finance using the yfinance package. The dataset consists of end of the day stock prices (16:00 ET) from "2012-05-18" to "2023-12-31" for a total of 2923 trading days.

3. Approaches

We used the framework set up in **DeepDow** to perform portfolio optimization. Core idea behind DeepDow framework is to treat the stock price data as a 3D tensor, with time, asset and channels as tensors. Here channels can be "Close", "Open", "High", "Low" and "Volume". We used "Close" as the only channel for most of our modeling. Using a rolling window DeepDow constructs smaller sets of feature and label tensors. These are used to optimize weightage of each stock in the portfolio based on a loss function. We used loss functions based on cumulative returns, Sharpe ratio and squared weights.

We used three different the models to perform the task. The first one is a custom model from the DeepDow package called BachelierNet. It combines the classical Markowitz framework with a deep learning network that estimates the expected returns. Our second model consists of a single LSTM cell used for all stocks in parallel, followed by a fully connected network and softmax allocator. This model directly estimates the portfolio weights. Our final model is very similar to the previous one, but instead of using a single LSTM for all the stocks, we used a different LSTM for each stock, and concatenated the results to feed them to the fully connected network.

4. Results

The trained models were tested on both the validation and test sets. Interested readers may find the results and plots in the Github: <https://github.com/OzkoOzzie/PortfolioOptimizationDL>

We observed the following in our studies.

1. Some stocks may perform well during the training and validation period but need to perform well during the testing period. This causes network to learn to weight these stocks heavily in the training period and causes a lag on the portfolio.
2. Requires skillful tuning of hyperparameters to get the best portfolio.
3. Considering efficiency of markets, extra data such as sentiment for a stock, as well as fundamental data may lead to better performance of the portfolio. One can add this as channel information in DeepDow to get a better model.