Deep Learning Portfolio Optimization

Authors: Tejus Setlur and Nigel Saurino
Supervisor: Dr. Daniel Totouom-Tangho

Abstract

The idea is to implement portfolio optimization with deep learning. We start off with building a deep learning portfolio optimization technique and we compare its performance with other portfolio optimization methods including Black-Litterman, mean-variance optimization and an all-weather portfolio. We researched CNNs and FCNs but ultimately chose to analyze an LSTM model's performance for our paper. We hope that this paper can offer an introduction to traditional portfolio optimization and an exploration of machine learning applications.

Introduction

Portfolio optimization is an essential component of a trading system. The optimisation aims to select the best asset distribution within a portfolio in order to maximize returns at a given risk level. This theory was pioneered in Markowitz's key work [4] and is widely known as modern portfolio theory (MPT). 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 adds extra risk to the portfolio and degrades subsequent performance. Further, the universe of available assets for constructing a portfolio is enormous. Taking the US stock markets as a single example, more than 5000 stocks are available to choose from. Indeed, a well rounded portfolio not only consists of stocks, but also is typically supplemented with bonds and commodities, further expanding the spectrum of choices.

We consider directly optimizing a portfolio, utilizing deep learning models. Unlike classical methods where expected returns are first predicted (typically through econometric models), we bypass this forecasting step to directly obtain asset allocations. Several works have shown that the return forecasting approach is not guaranteed to maximize the performance of a portfolio, as the prediction steps attempt to minimize a prediction loss which is not the overall reward from the portfolio. In contrast, our approach is to directly optimize the Sharpe ratio, thus maximizing return per unit of risk. Our framework starts by concatenating multiple features from different assets to form a single observation and then uses a neural network to extract salient information and output portfolio weights so as to maximize the Sharpe ratio.

As a benchmark, we chose to use the Black-Litterman method for portfolio optimization. This approach allows managers to layer in their subjective investment views alongside the optimal portfolio weights that are calculated through a traditional mean-variance optimization. Our paper addresses the implementation, advantages and challenges of using Black-Litterman in practice (including proper selection of a manager's views). Overall, our analysis of Black-Litterman offered us an opportunity to dive even deeper into the material after learning about it in class this semester.

Instead of choosing individual assets, Exchange-Traded Funds (ETFs) of market indices are selected to form a portfolio. We use four market indices: US total stock index (VTI), US aggregate bond index (AGG), US commodity index (DBC) and Volatility Index (VIX). All of these indices are popularly traded ETFs that offer high liquidity and relatively small expense ratios. Trading indices substantially reduces the possible universe of asset choices and gains exposure to most securities. Further, these indices are generally uncorrelated, or even negatively correlated. Individual instruments in the same asset class, however, often exhibit strong positive correlations.

Methodology

In this section we discuss the different models that we have used. We pay special attention to the architecture, assumptions, math and advantages of each model.

Deep Learning Model for Portfolio Optimization

The Sharpe ratio compares the return of an investment with its risk. It's a mathematical expression of the insight that excess returns over a period of time may signify more volatility and risk, rather than investing skill. 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):

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

where $E(R_p)$ and $Std(R_p)$ are the estimates of the mean and standard deviation of portfolio returns. Specifically, for a trading period of t = {1, · · · , T}, we can maximize the following objective function:

$$L_T = \frac{E(R_{p,t})}{\sqrt{E(R^2_{p,t}) - (E(R_{p,t}))^2}}$$

$$E(R_{p,t}) = \frac{1}{T}\sum_{i=1}^{T} R_{p,t}$$

where $R_{p,t}$ is realized portfolio return over n assets at time t denoted as:

$$R_{p,t} = \sum_{i=1}^{n} w_{i,t-1} \cdot r_{i,t}$$

where $r_{i,t}$ is the return of asset i with $r_{i,t} = (p_{i,t}/ p_{i,t-1} - 1)$. We represent the allocation ratio (position) of asset i as $w_{i,t} \in [0, 1]$ and $\sum_{i}^{n} w_{i,t} = 1$. In our approach, a neural network f with parameters θ is adopted to the model $w_{i,t}$ for a long only portfolio:

$$w_{i,t} = f(\theta | x_t)$$

where x_t represents the current market information and we bypass the classical forecasting step by linking the inputs with positions to maximize the Sharpe over trading period T, namely L_T . However, a long-only portfolio imposes constraints that require weights to be positive and summed to one, we use softmax outputs to fulfill these requirements:

$$w_{i,t} = \frac{exp(w`_{i,t})}{\sum_{j}^{n} exp(w`_{j,t})}$$, where $w`_{i,t}$ are the new raw weights.

Such a framework can be optimized using unconstrained optimization methods. Particularly, we use gradient ascent to maximize the Sharpe ratio.
We take the derivative of L_T using the chain rule:

$$\frac{dL_T}{dw} = \frac{d}{dw}\left\{\frac{A}{\sqrt{B - A^2}}\right\} = \frac{dL_T}{dA} \cdot \frac{dA}{dw} + \frac{dL_T}{dB} \cdot \frac{dB}{dw}$$

$$= \sum_{t=1}^{T}\left\{\frac{dL_T}{dA} \cdot \frac{dA}{dR_{p,t}} + \frac{dL_T}{dB} \cdot \frac{dB}{dR_{p,t}}\right\} \cdot \frac{dL_T}{dw}$$

$$= \sum_{t=1}^{T}\left\{\frac{dL_T}{dA} \cdot \frac{dA}{dR_{p,t}} + \frac{dL_T}{dB} \cdot \frac{dB}{dR_{p,t}}\right\} \cdot \left\{\frac{dL_T}{dF_t} \cdot \frac{dF_t}{dw} + \frac{dL_T}{dF_{t-1}} \cdot \frac{dF_{t-1}}{dw}\right\}$$

where F_t is the holdings of each of the ETFs of the portfolio at time *t.*

The necessary partial derivatives of the return function are:

$$\frac{dL_t}{dF_t} = \frac{d}{dF_t}\left\{F_{t-1} \cdot r_t - \delta|F_t - F_{t-1}|\right\}$$
$$= \frac{d}{dF_t}\left\{- \delta \cdot |F_t - F_{t-1}|\right\} = - \delta \cdot sgn(F_t - F_{t-1})$$
$$\frac{dL_T}{dF_{t-1}} = \frac{d}{dF_{t-1}}\left\{F_{t-1} \cdot r_t - \delta \cdot |F_t - F_{t-1}|\right\}$$
$$= r_t - \frac{d}{dF_{t-1}}\left\{- \delta \cdot |F_t - F_{t-1}|\right\} = r_t + \delta \cdot sgn(F_t - F_{t-1})$$

Then, the partial derivatives dF_t /dw and dF_{t-1} /dw must be calculated:

$$\frac{dF_t}{dw} = \frac{d}{dw}\{tanh(w^T x_t)^2\} = (1 - tanh(w^T x_t)^2) \cdot \frac{d}{dw}\{w^T x_t\}$$
$$= (1 - tanh(w^T x_t)^2) \cdot \left\{x_t + w_{M+2}\frac{dF_{t-1}}{dw}\right\}$$

Note that the derivative dF_t/dw is recurrent and depends on all previous values of dF_t/dw.

Once we obtain *∂LT/∂θ*, we can repeatedly compute this value from training data and update the parameters by using gradient ascent:

$$\theta_{new} = \theta_{old} + \alpha\frac{\partial L_T}{\partial \theta}$$

where α is the learning rate and the process can be repeated for many epochs until the convergence of Sharpe ratio or the optimization of validation performance is achieved.

Black-Litterman for Portfolio Optimization

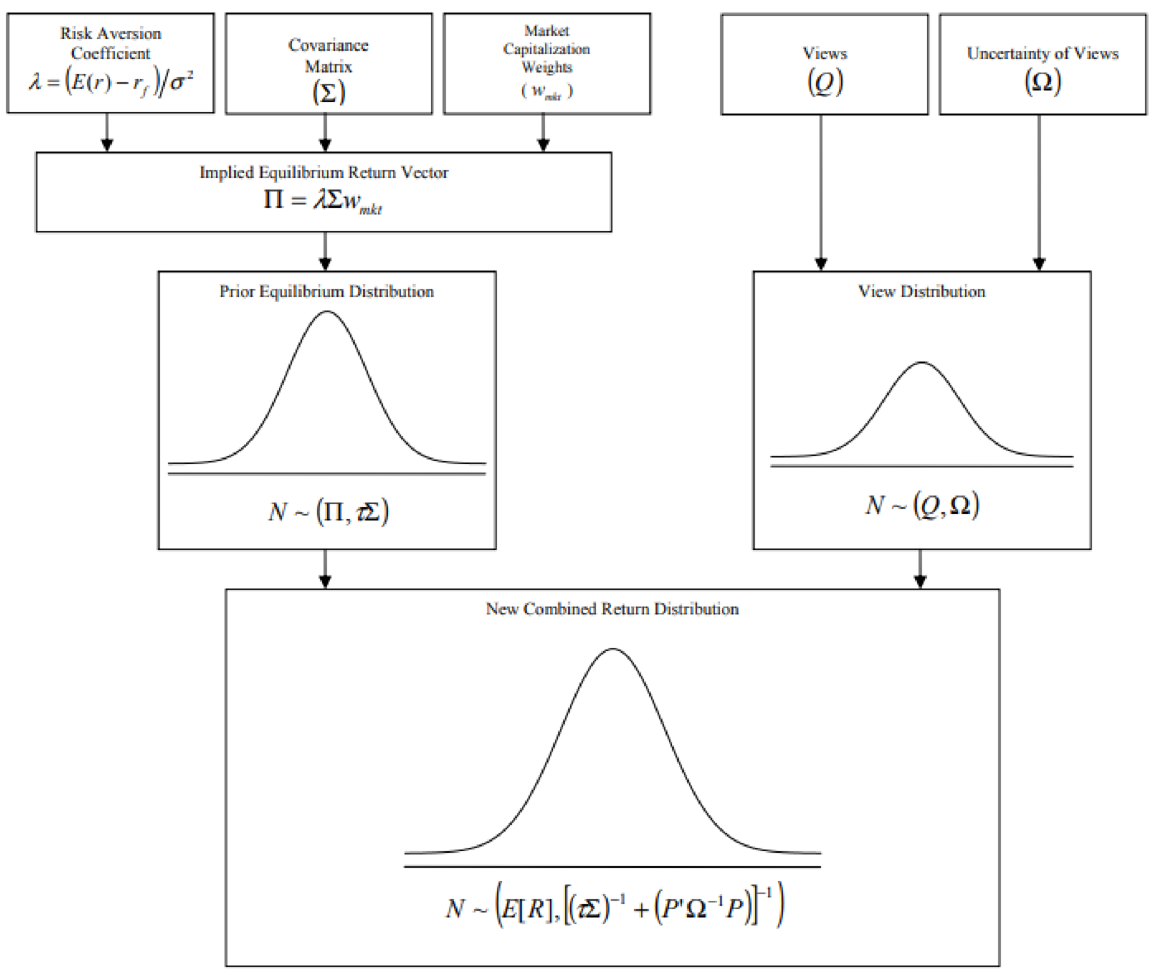
The Black-Litterman model is an extension of Modern Portfolio Theory first introduced by the economist Dr. Harry Markowitz. This model is an adaptation of the classic mean-variance framework which enables investors to combine their unique views regarding the performance of various assets with the market equilibrium in a manner that results in intuitive, diversified portfolios. As an extension of the traditional mean-variance model, Black-Litterman forecasts expected returns, incorporates an investor’s subjective views and confidence level, and then calculates portfolio allocations from the mixed estimate of expected returns (“posterior distribution”). In short, the steps for the Black-Litterman model are outlined below:

1) Step 1: Find Risk Premia to demand from stocks in market portfolio
2) Step 2: Adjust Risk Premia of individual stocks with our own views

3) Step 3: Have mixed estimate of forward-looking returns
4) Step 4: Perform classic mean-variance optimization to calculate weights for individual stocks in portfolio using the covariance matrix
5) Step 5: End result is the set of expected returns of individual assets in the portfolio and their associated optimal portfolio weights

As a visual, we are providing an excerpt from Dr. Daniel Totouom's lectures below which should be referenced as we discuss some of the key mathematical principles in this section.

Figure 1 Deriving the New Combined Return Vector ($E[R]$)



* *The variance of the New Combined Return Distribution is derived in Satchell and Scowcroft (2000).*

The mathematics underlying the Black-Litterman framework are similar to a mean-variance optimization with a few notable exceptions.

An Introduction to the Mathematics of the Black-Litterman Model

- Key Variables:
  - τ= Scalar (0.025; highly dependent on practitioner)
  - Π = 4x1; vector of the expected excess returns
  - Σ = 4x4; covariance matrix representing the uncertainty of the expected excess returns
  - Q= 2x1; views vector
  - P = 2x4; link matrix
  - Ω = 2x2; uncertainty matrix of our views
- Rather than solving for optimal rates, Black-Litterman solves for the implied equilibrium excess returns. First, start with the investor's utility function, then take the derivative and solve for Π:

$$Investor\ Utility\ = w^T * \Pi - \frac{1}{2} * A * w^T * \Sigma * w$$

$$\frac{du}{dw} = \Pi - \frac{1}{2} * 2 * A * \Sigma * w = \Pi - A * \Sigma * w$$

$$\Pi = A * \Sigma * w$$

- Where:
  - A is the risk aversion measure ("price of risk") and is equivalent to $\frac{Excess\ return\ on\ the\ market}{Variance\ of\ the\ market} = \frac{E[r_m] - r_f}{\sigma_m^2}$
  - Σ is the covariance matrix of our portfolio
  - w is the weight vector of the portfolio
- The real breakthrough in the Black-Litterman approach was achieved by incorporating subjective investor views into the mean-variance optimization method. In practice, it is more common for practitioners to exercise relative views between assets of a portfolio. However, in our paper, we only consider absolute expectations of an assets performance for the following period.
- Σ^{-1} is the measure of confidence of our estimated implied equilibrium excess returns
- Black-Littermamn is able to define the uncertainty of our views as a combination of a scalar τ, the link matrix P, the covariance matrix Σ and the transpose of the link matrix P^T:

$$\Omega = \tau * P * \Sigma * P^T$$

- Therefore, our confidence in our views is simply Ω^{-1}
- Ultimately, the Black-Litterman model gives us an estimate of our excess returns updated for our investor views by performing a weighted average of the implied

equilibrium excess returns (Π) and our views (Q). Our weights for the function will be our confidence of Π = $(\tau\Sigma)^{-1}$ and the confidence in Q times our link matrix to indicate which assets the views are about $(P^T * \Omega^{-1})$. Together,

$$(\tau\Sigma)^{-1} * \ \Pi \ + (P^T * \Omega^{-1}) * Q$$

- You will recognize this as the second term of the Black-Litterman formula. The first term of the formula is the inverse of the sum of the weights and is included to confirm that the sum of the weights that we derive is equal to 1, ensuring that the target portfolio is not over/underallocated. In summary,

$$E[r_p] \ - \ r_r = [(\tau\Sigma)^{-1} + (P^T * \Omega^{-1} * \ P)] * [(\tau\Sigma)^{-1} * \ \Pi \ + (P^T * \Omega^{-1}) * Q]$$

Architecture

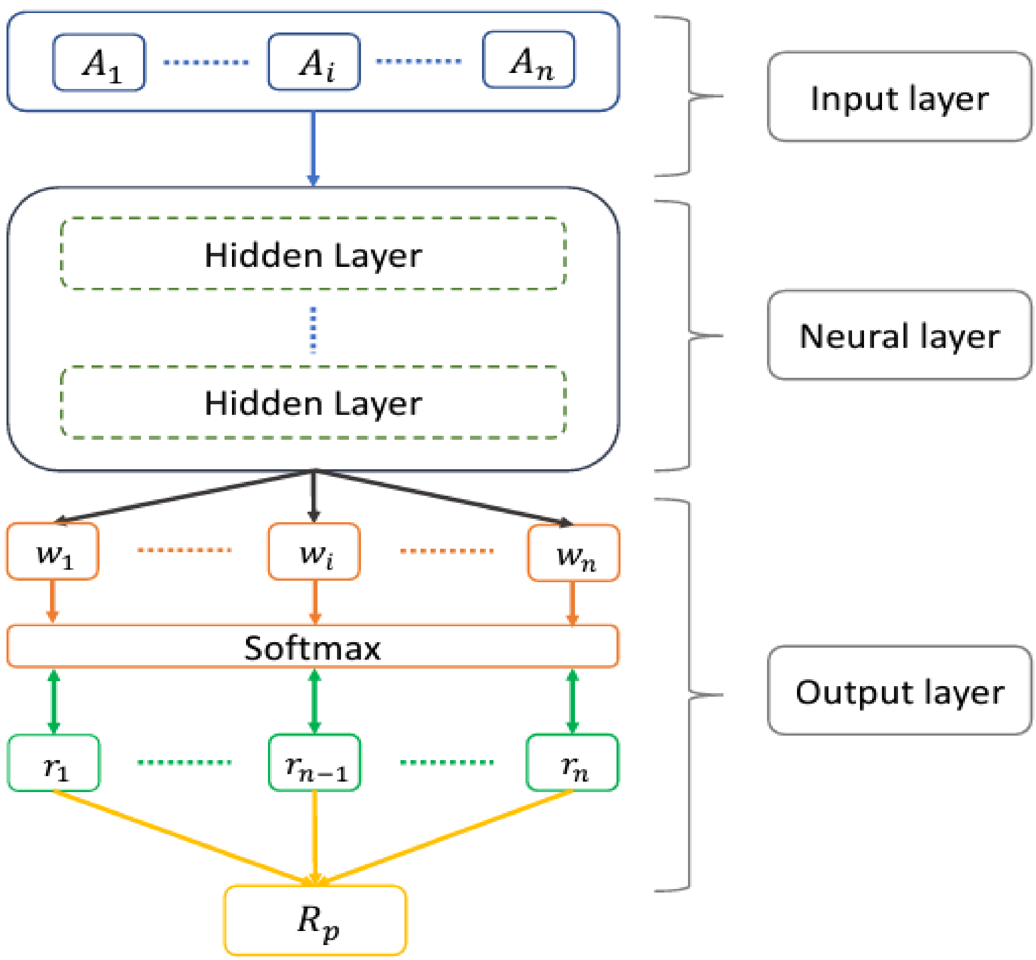
We depict our network architecture in Figure 2. Our model consists of three main building blocks: input layer, neural layer and output layer. The idea of this design is to use neural networks to extract cross-sectional features from input assets. Features extracted from deep learning models have been suggested to perform better than traditional hand-crafted features [5]. Once features have been extracted, the model outputs portfolio weights and we obtain realized returns to maximize Sharpe ratio.

Input layer: We denote each asset as Ai and we have n assets to form a portfolio. A single input is prepared by concatenating information from all assets. For example, the input features of one asset can be its past prices and returns with a dimension of (k, 2) where k represents the lookback window. By stacking features across all assets, the dimension of the resulting input would be (k, 2 × n). We can then feed this input to the network and expect non-linear features being extracted.

Neural layer: A series of hidden layers can be stacked to form a network, however, in practice, this part requires lots of experiments as there are plentiful ways of combining hidden layers and the performance often depends on the design of architecture. We have tested deep learning models including fully connected neural network (FCN), convolutional neural network (CNN) and Long Short-Term Memory (LSTM) [9]. Overall, LSTMs deliver the best performance for modeling daily financial data and a number of works [6, 7, 8] support this observation.
We note the problem of FCN is its problem of severe overfitting. As it assigns parameters to each input feature, this results in an excess number of parameters. The LSTM operates with a cell structure that has gate mechanisms to summarize and filter information from its long history, so the model ends up with fewer trainable parameters and achieves better generalization results. In contrast, CNNs with a strong smoothing

(typical of large convolutional filters) tend to have underfitting problems, such that over smooth solutions are obtained. Due to the design of parameter sharing and the convolution operations, we experience CNNs to over filter the inputs. However, we note that CNNs appear to be excellent candidates for modeling high-frequency financial data such as limit order books [10].

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 (w1, · · · , wn) is equal to the number of assets in our portfolio, and we can multiply these portfolio weights with associated assets' returns (r1, · · · , rn) to calculate realized portfolio returns (Rp). Once realized 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.



Model architecture schematic. Overall, our model contains three main building blocks: input layer, neural layer and output layer.

Empirical Results

Data

Our analysis considered four different exchange traded funds (ETFs) which allowed us to capture the behavior of a range of markets in a diversified manner without selecting specific underlying securities as proxies for their overall sector. An exchange-traded fund (ETF) is a type of pooled investment security that operates much like a mutual fund. Typically, ETFs will track a particular index, sector, commodity, or other assets, but unlike mutual funds, ETFs can be purchased or sold on a stock exchange the same way that a regular stock can. We have selected these ETFs on the basis of their longevity and liquidity. Each ETF is well-known in industry and has been around for over a decade. To create a diversified portfolio that is representative of key asset classes from financial markets, we used the Vanguard Total Index (VTI), the US aggregate bond index (AGG), US commodity index (DBC) and the infamous "fear gauge", Volatility Index (VIX). Diversification is a key to strong sharpe ratios and so we wanted to avoid any structural headwinds by having low correlation asset classes. We evaluated each model's performance over the 2011 - 2020 period using the sharpe ratio as our objective function. Additionally, for our deep learning model, we used data from 2006 - 2011 as a training set. The 2010s saw regime change and several market shocks including the taper tantrum, a low volatility environment, rising interest rates and a global pandemic. As a result, we feel confident with the robustness of the model performance over our test set's timeline.

At the professor's suggestion, we decided to implement the Black-Litterman model in python as a benchmark for our deep learning model's performance over the same test set. We chose to impose a daily rebalancing of the portfolio using a rolling window of 50 days to estimate returns and the covariance matrix "Σ". Our weights are calculated to maximize the sharpe ratio and offer a consistent framework to evaluate relative performance for the models.

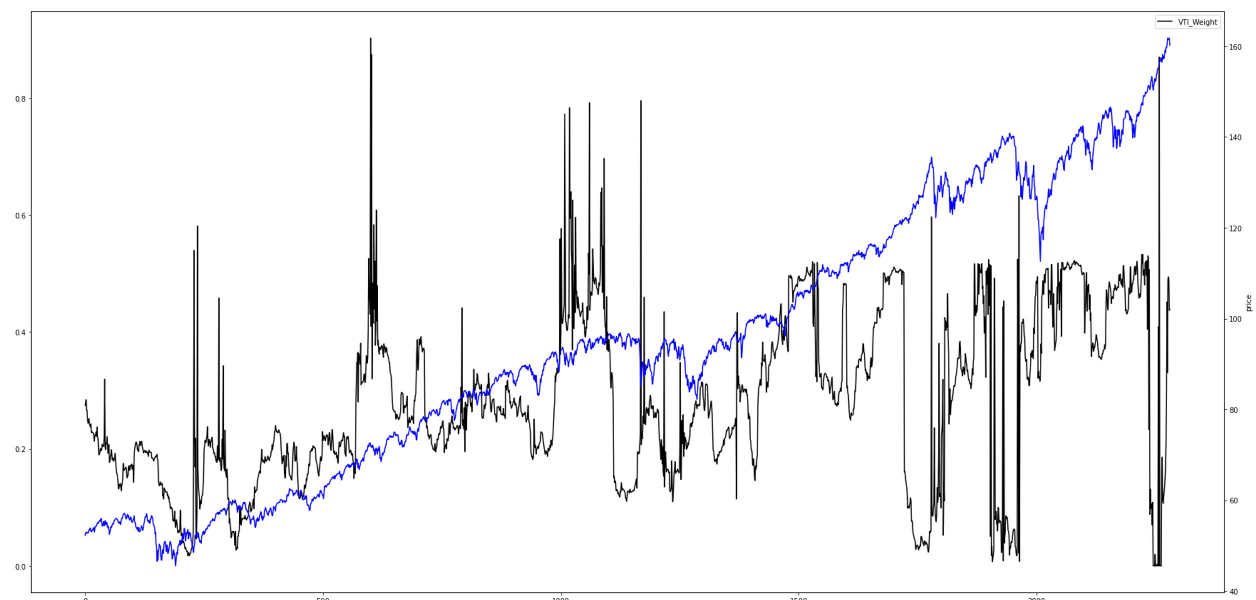
With Black-Litterman, an advantage clearly lies in the investor's ability to tilt a portfolio towards their views. However, this emphasizes the method with which the investor arrives at these views. In our first implementation, we approached the views selection naively with a fixed expectation for outperformance/underperformance depending on the asset class. However, the professor challenged us to consider more informed approaches to view selection during the presentation.

As a result, we set three different methodologies that could be switched between and tuned by the user. The first method emphasized the impact of the assets with the minimum and maximum returns by taking an absolute view that increased the top performing asset's expected return and decreased the worst performing asset's expected return in the vector. The second methodology performed the opposite and instead decreased the expected return of the top performing asset and increased the expected return of the worst performing asset. The final methodology applied fixed absolute views across the assets. For all data and visualizations below, we referenced the first methodology.

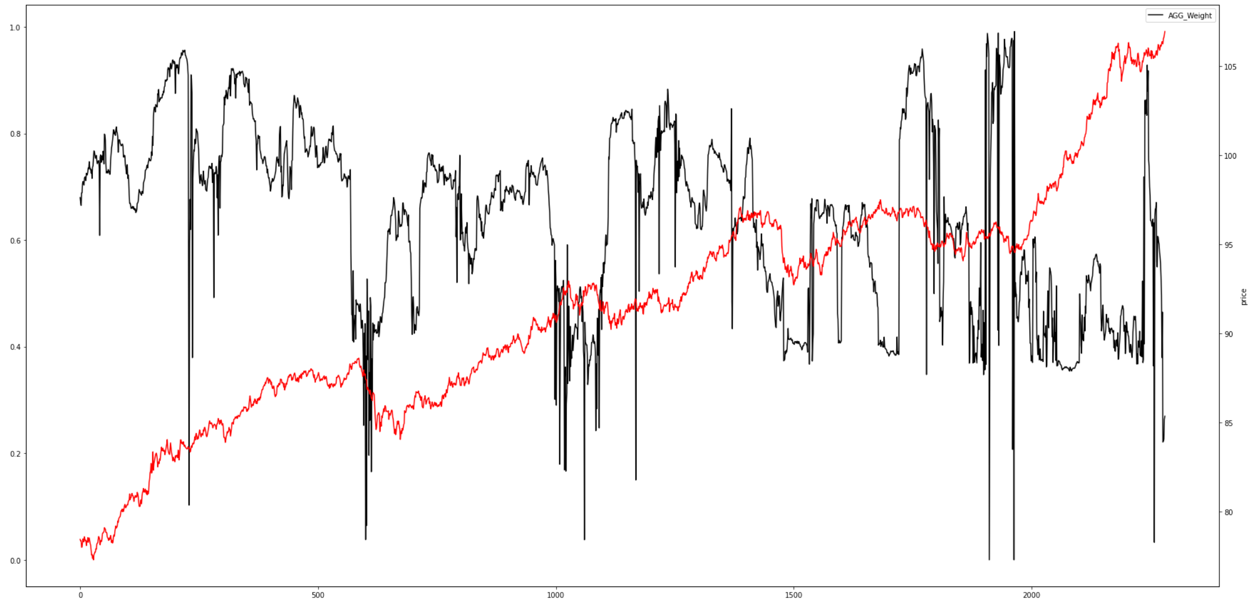
After further investigation, we think it would be extremely interesting to incorporate a factor approach (specifically momentum) in order to extend the paper's analysis. In our research for view formulation after the presentation, we came across a few great papers from AQR that provided an introduction to factor investing along with their own results (Asness et al. [2] & Ross et al. [3]). It would require a much deeper dive into factor investing but would be interesting to see how the Black-Litterman results are impacted by certain investing styles.

After working through the models, we thought it would be interesting to include a few visualizations to help us understand how Black-Litterman's allocations change over time and how the final allocations relate to the Deep Learning model's calculated weights.

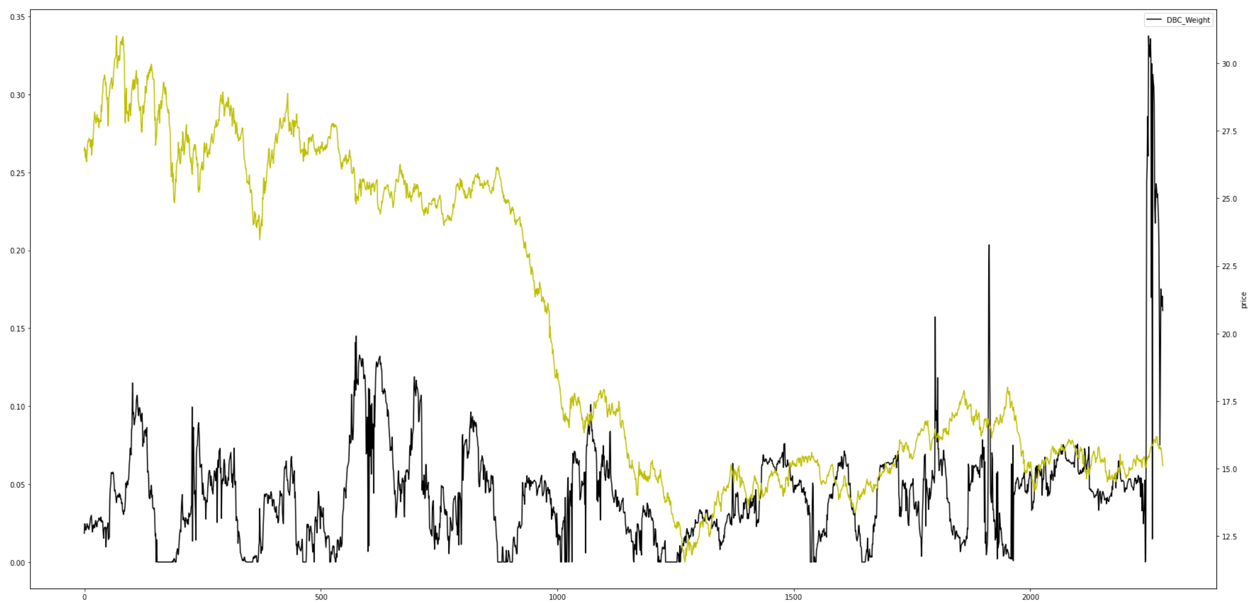
VTI Price vs VTI Weight (In Portfolio)



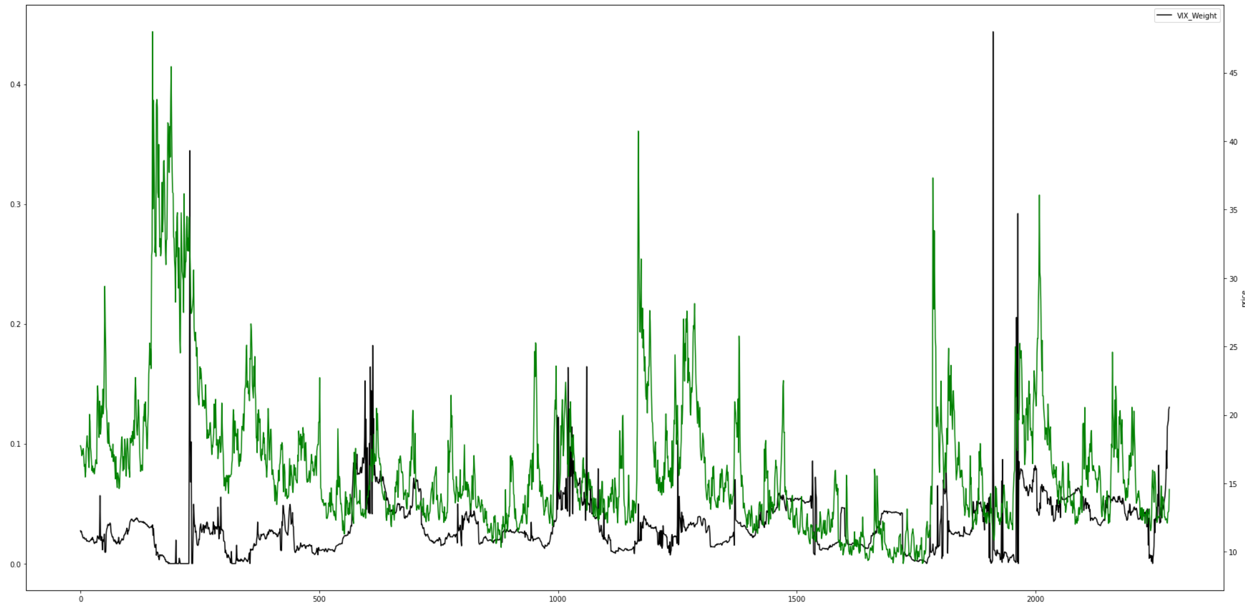
AGG Price vs AGG Weight (In Portfolio)



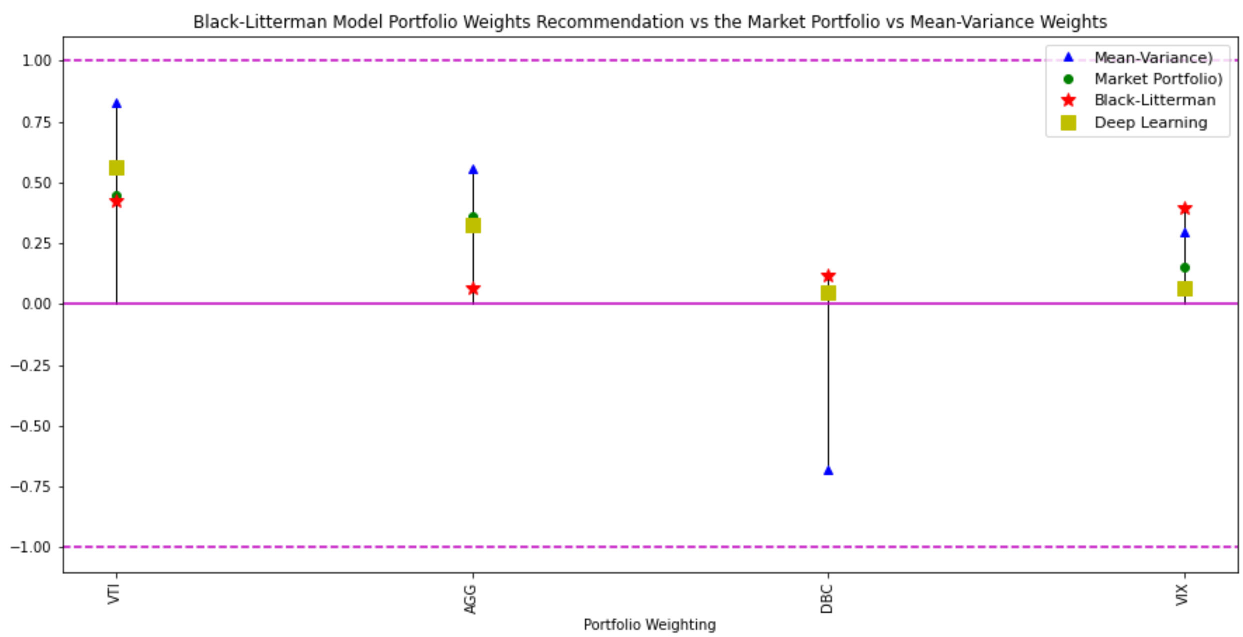
DBC Price vs DBC Weight (In Portfolio)



VIX Price vs VIX Weight (In Portfolio)



The final allocations for various portfolios have:



The Sharpe ratio is a classic industry metric and offers important implications from a single value. In our analysis, you can see that the ratios of each allocation strategy have similar performance results for the years between 2011 and 2020.

While we ran out of time to implement an allocation-based, all-weather portfolio, we thought it would still be instructive to include the results from our reference paper (Zhang et al. [1]). The table of Sharpe ratios by strategy has been included below:

Strategy	**E(R)**	**Std(R)**	**Sharpe**
All-weather	0.282	0.303	0.929
Mean-Variance	0.082	0.108	0.759
Black-Litterman	0.082	0.054	1.501
Deep Learning	0.130	0.071	1.833
Deep Learning (out of sample - suggested during presentation)(2020-2022)	0.001	0.106	0.921

All of these metrics are annualized. As an extension of our work, we would consider implementing a range of performance metrics including downside deviation of return (DD(R)), the Sortino ratio, Calmar ratio, maximum drawdown (MDD), percentage of positive return (% of + Ret) and the ratio of positive to negative return (AVG P / AVG L).

Conclusion

In the course of this project we have implemented different portfolio optimization strategies such as the All-weather, Mean-Variance, Black-Litterman, and Deep Learning strategy. We consider sharpe ratio as a measure of efficacy of each strategy as it quantifies the returns per unit risk. When we compare the Sharpe ratio output we realize that the deep learning technique provides the optimal portfolio. The deep learning model uses 20 epochs for the data we have used to converse on the optimal solution but this could differ for future work. The in-sample output of Deep learning technique pips the Black Litterman benchmark technique we implemented suggesting that the scope of deep learning in the field of portfolio optimization is increasing. As suggested during the presentation, we implemented the out of sample optimization using the trained model and it converged with weight distribution of Asset List: ['VTI': 0.37155417, 'AGG': 0.10824852, 'DBC': 0.42225987, 'VIX': 0.09793751]. We explain the observed dip in the out of sample sharpe ratio by indicating that 2020 was a year of slow growth due to the COVID-19 pandemic and the returns could not have been great.

References

[1] https://arxiv.org/pdf/2005.13665.pdf
[2] Fact, Fiction and Momentum Investing
[3] Implementing Momentum: What Have We Learned?
[4] Harry Markowitz. Portfolio selection. The journal of finance, 7(1):77–91, 1952.
[5] Zihao Zhang, Stefan Zohren, and Roberts Stephen. Deep reinforcement learning for trading. The Journal of Financial Data Science, 2020.
[6] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning to detect price change indications in financial markets. In 2017 25th European Signal Processing Conference (EUSIPCO), pages 2511–2515. IEEE, 2017.
[7] Bryan Lim, Stefan Zohren, and Stephen Roberts. Enhancing time-series momentum strategies using deep neural networks. The Journal of Financial Data Science, 1(4):19–38, 2019.
[8] Zihao Zhang, Stefan Zohren, and Roberts Stephen. Deep reinforcement learning for trading. The Journal of Financial Data Science, 2020.
[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.
[10] Zihao Zhang, Stefan Zohren, and Stephen Roberts. DeepLOB: Deep convolutional neural networks for limit order books. IEEE Transactions on Signal Processing, 67(11):3001–3012, 2019.