**Section 1. Problem Description**.

In the field of Systematic Finance, investors utilize a variety of signals to inform stock market predictions and drive investment decisions. These signals can be derived from a multitude of sources, including market data (e.g., stock prices, volumes, and volatility), technical indicators (e.g., moving averages, momentum indicators), fundamental data (e.g., earnings reports, economic indicators) and alternative data. Once these individual signals are collected, they are typically aggregated to form a more comprehensive signal, which in turn guides investment decisions.

Each of these signals offers a distinct perspective on market behavior. However, no single signal is universally predictive, and their relevance can vary significantly through time. As a result, signal aggregation, which is the process of combining multiple signals into a unified score to guide portfolio decisions, has become a key problem in quantitative finance.

A commonly used approach to signal aggregation is the uniform weight strategy, also known as the 1/n method, where each signal is given an equal importance regardless of their individual predictive power. While it’s simple to implement, robust and adjust well to drastic change, it assumes that all signals are equally valuable or relevant in different market conditions, which is rarely the case in practice. As a result, investors may end up treating less relevant signals the same as those that are more predictive, which can lead to suboptimal investment decisions and reduced returns.

Another alternative is expert-based weighting, where domain experts are manually responsible for assigning weights to different signals. However, this method suffers from subjectivity of these experts, lacks scalability, as determining the relative importance of each signal becomes increasingly difficult as the number of signals grows, and often fails to adapt quickly to drastic market change. Moreover, this approach lacks data-driven support, which is a core principle of systematic trading. In systematic trading, the goal is to rely on historical data to inform decision-making, ensuring that the model adapts to market conditions rather than relying on subjective or static assumptions.

The challenge, therefore, lies in the development of a data-driven, adaptive method for signal aggregation that can:

* Automatically determine the relevance of each signal for each asset
* Adapt to changing market conditions over time
* Improve the portfolio-level decision-making and returns
* Execute efficiently within reasonable time constraints, allowing for timely portfolio decisions in practical settings (daily or monthly rebalancing), even as the number of signals or assets grows.

Within this context, we aim to learn the optimal weight matrix *W* for combining multiple signals in a way that maximizes the return of a portfolio. Let:

* denote the score provided by signal i for asset k
* denote the weight assigned to the signal i
* denote the portfolio weight allocated to asset k, which is computed as a function of the aggregated score across all signals.

The aggregated score for the asset k is:

This aggregated score is then used to compute the portfolio allocation vector , subject to:

Hence, the core optimization objective is to find the signal weights 𝑤 such that the resulting portfolio achieves maximum expected return over historical data. In this problem, the action of choosing signal weights leads to a downstream investment decision, which produces a measurable reward being the annualized or monthly portfolio return.

The goal is to learn this signal weighting scheme from historical data in a way that generalizes well to future periods. This formulation reflects a broader class of problems where learning and optimization are coupled, and decision-making is informed by feedback from actual performance.

**Section 2. Methodology**

In this section, we describe the methodology that was used to develop, train, and evaluate the data-driven approach for portfolio optimization that was detailed in the previous section. The overarching goal is to improve investment decision-making by aggregating multiple technical signals using a learning algorithm in a way that leads to greater portfolio returns, compared to the naïve 1/n heuristic method.

**2.1 Problem Setting**

For this project, we restricted our universe to a pool of 20 companies selected from the S&P 500 index, that represents the 500 largest U.S. companies in the stock market. For each of the trading day, we calculate a score for each company based on a set of three preselected momentum-based indicators:

* Simple Moving Average (SMA): Measure trend direction on the last year
* Relative Strength Index (RSI): Captures the overbought/oversold conditions over the last 14 days
* Moving Average Convergence/Divergence (MACD): Indicates momentum changes in the last month

These signals are commonly used in finance because they are computationally efficient and are great representations of a stock overall health. Hence, for each trading day and for each of the 20 stocks, we get 3 signal scores. The general idea is to learn a set of weights (one per signal) that will reflect the relative value of the signals in making portfolio-level decisions. We then use these weights to compute a combined signal score (alpha signal score) per stock, which guide the portfolio allocation decision.

Formally, we have that:

= number of signals in the solution (in this case, 3)

= number of stocks in the solution (in this case, 20)

= weight of the signal

= score of the signal for the stock

= the percentage of the portfolio allocated to stock

Hence, the aggregate score for each stock is:

We then define the optimization problem to determine the portfolio allocation that maximizes a combination of the aggregated signal strength and portfolio-level return, while considering variance.

**2.2 Optimization-Base Portfolio Construction**

We explored three distinct optimization solvers for translating the alpha signal scores into the portfolio allocation:

1) Mean-Variance Optimization (MVO)

The first optimization solution that we used is the Mean-Variance Optimization, which is a solver using historical return data. This is usually good to deal with the volatility of the stock market but doesn’t take into account our signal scores. Here is the classic MVO formulation:

subject to:

where is the expected return vector, is the covariance matrix, and is our coefficient of risk-aversion.

2) Signal-Score Strategy

This is a simple data-driven strategy that assigns weights based purely on the aggregated alpha signal scores per stock. That is; after computing our , we normalize use only the scores to make our optimization decisions. This strategy is fast, interpretable and considers our signal scores, but lacks to consider volatility in the stock market.

3) Signal-Weighted MVO (Quadratic Programming)

Finally, we propose a mixture of the two formulations that combines the signal-based score and the volatility of the market by including both previous methods. Hence, the optimization problem becomes the following:

subject to:

This is a quadratic programming problem (QP), that can be solved using cvxopt. We also tested an implementation of a max allocation constraint of 20%, to include a diversification enforcement in the solver, which adds another constraint:

**2.3 Learning the Signal Weights**

The weights of are not fixed in advance but instead are learned through our architecture to optimize downstream portfolio performance. We use two learning paradigms that we saw in class:

1. **Sequential Learning and Optimization (SLO) via Linear Model**

As a first approach, we implemented a sequential learning and optimization architecture, where a linear model is trained to find the optimal signal weights . To validate this method, we started by generating simulated signals and returns, which allowed us to get an idea as to if the learning process could effectively learn an optimal weight vector in a controlled environment before applying it to real market data.

We were interested in evaluating how much data we require to learn the signal weight matrix. Our simulation results revealed that with only a single year of trading dates, we can learn a stable linear weight matrix. This finding is highly relevant because it suggests that our signal aggregation model can adapt rapidly to market dynamics and doesn’t require an extensive historical dataset to have some practical results.

Once we validate, we used the method on real data, that is, the RSI, SMA and MACD scores on the 20 S&P500 stocks using the historical signal values to predict subsequent portfolio returns under different weight configurations. The loss function penalizes poor portfolio performance, and the model learns to adjust the weights accordingly.

1. **Neural Network training with surrogate softmax loss**

We also explored an alternative approach to learning based on neural network trained with a surrogate softmax loss function to learn the signal weights.

The idea behind this approach even though the linear model might offer simplicity and interpretability, it may not fully capture the complex relationship between signals and portfolio returns. A neural network here provides a more flexible function approximator, that allows the model to discover richer patterns in how signals should be aggregated.

However, directly optimizing portfolio returns during training is challenging because the function return is often non-differentiable or highly irregular. Hence, to address this, we used a surrogate loss function based on the softmax operator applied over the asset scores.

Overall, the neural network with surrogate loss pipeline goes as follows. We start by downloading the data and computing the scores, just like in the linear approach and we feed these scores for all three signals as input features for the model.

The neural network receives these signal scores and outputs a set of learned signal weights that are used to compute the aggregated score for each asset. After calculating the aggregated scored, we apply a softmax transformation:

Where can be interpreted as a “soft” portfolio weight assigned to asset k.

We use the softmax operator for multiple reasons. First, the softmax allows us to enforce the natural constraints of the portfolio weights being nonnegative and summing to 1 in a differentiable way. Secondly, it allows for a smooth portfolio allocation landscape, that makes the optimization more stable and tractable for gradient-based learning. Finally, the softmax provides a continuous approximation to the discrete portfolio allocation decision, allowing the use of standard backpropagation techniques.

Hence, the surrogate loss is defined as the negative expected portfolio return from the softmax-transformed weights:

Where is the realized return of the asset k over the evaluation period of 252 trading days (a year)

By minimizing the surrogate loss, the neural network learns the signal weights that indirectly maximize the expected portfolio return, while ensuring that the learning process remains differentiable and stable.

This approach allows the model to learn the more complex nonlinear relationships between the signals and portfolio returns, at the cost of increased model complexity and greater risk of overfitting due to the limited availability of the historical training data.

**2.4 Practical constraints**

When developing the pipeline, we considered some practical constraints relevant to the real-world application deployments:

* The system must be fast enough to run in a single day as we need to make decisions on the same day as we obtain the data.
* We need to find a model that has a right balance of complexity and speed, preferring interpretable, light models that generalize well unseen conditions.

**2.5 Goal of the methodology**

All methods and adaptation are ultimately evaluated against the 1/n strategy to evaluate whether learning signal weights leads to statistically and practically significant improvements in portfolio returns. The central hypothesis behind is that not all signals are equally useful, and learning to weigh them adaptively leads to better portfolio outcomes.

**Section 3. Discussion**

The portfolio allocation problem explored in this project fits into the “Predict-then-Optimize” and “Learning to Optimize” categories of the machine learning and optimization taxonomy that was discussed in class. In the Predict-then-Optimize paradigm, machine learning is used to estimate parameters, which are, in this case signal weights, that are then passed into an optimization problem to generate a portfolio. However, this project can also align itself closely with the Integrated Learning and Optimization, as the signal weights can be learned through their impact on the portfolio objective by directly optimizing over the outcome, which is what we are doing in this project when training on the returns. Additionally, the use of surrogate loss plays a key role in the optimization process, particularly in enabling the model to be trained on returns. This concept of non-differentiable optimization was presented in the course material and is critical in this project to ensure that the portfolio returns can be optimized in a stable and differentiable manner, allowing the use of gradient-based learning techniques.

This direct optimization of decision quality, instead of predictive accuracy, represents the shift from traditional machine learning to decision-focused learning. This is especially well suited for financial problems where small improvements in decision quality can translate into significant real-world benefits, and where learning from historical data allows adaptation to changing complex market dynamics. Furthermore, these hybrid techniques address scalability challenges that can be found in these types of problem by reducing the need to explicitly model high-dimensional interactions or retrain large forecasting models, which are extremely computationally expensive with large portfolios or many signals.

One of the main strengths of this methodology is its adaptability to changing market conditions and its ability to integrate a lot of different sources of signals into a single unified framework. By optimizing directly from the portfolio returns, we avoid the common pitfall of maximizing predictive accuracy at the cost of poor decisions. Additionally, this method is easy to deploy, which is very useful in a pragmatic way.

However, there are multiple limitations that need to be addressed. First, we have that the performance of the model can be very sensitive to the quality and availability of the weighted signals, meaning that if input signals or are noisy or poorly constructed or the weight matrix isn’t trained well, the optimization can inherit that weakness. Second, we have that this architecture takes time to adapt to a drastic sudden change into the stock market. If the training dates do not reflect the current stock market on a date, well they could have an even worse return than the current 1/n method. Furthermore, the transaction costs and market impact, which are critical to in real-world settings, are not considered in this formulation of the problem here.

While financial portfolio optimization itself may not seem to directly relate to sustainable development at first glance, this methodology offers several meaningful benefits on sustainable development. By integrating ESG (Environmental, Social and Governance) indicators into the signal generation or weighting process, this optimization framework could be used to prioritize investments in compagnies with stronger sustainability profiles. This enables the design of portfolios that align with not just return and risk goals, but also sustainable and ethical goals, which would allow for individuals or companies that want to prioritize ESG to invest, knowing their values are being considered in the equation. We can also easily implement constraints that stop our optimizer from allocating resources to companies that have bad sustainability profiles. Overall, this could directly support the redirection of capital toward the businesses that contribute positively to climate, labor practices, and corporate governance.

As sustainable investing becomes more and more mainstream, these tools would optimize returns while honoring ESG constraints and can empower institutions and individuals to make more responsible financial decisions.

**Section 4: Question and Hypothesis**

**Question**:

Can a portfolio optimization model that learns to assign optimal weights to different signals outperform the traditional 1/n portfolio allocation strategy in terms of return and risk-adjusted performance?

**Hypothesis**:

I hypothesize that the portfolio optimization model, which learns a weighted combination of technical indicators (such as RSI, MACD, and SMA) to score assets, will outperform the traditional 1/n portfolio allocation approach. Unlike the traditional 1/n approach, that treats all signals equally, the weighted signal-based model uses learning to determine how much importance should be given to each signal. By learning which signals are most relevant to future returns, and dynamically adjusting asset allocations based on these weighted signal scores, the architecture is expected to provide a higher return. Moreover, this method can effectively reduce the influence of noisy or irrelevant signals by assigning their weighted value to a very low weight. This signal selection mechanism enhances the portfolio efficiency and ensures that it only considers the most informative signals when driving investment decisions.

Additionally, I hypothesize that the neural network approach with the surrogate loss function should further improve on the performance. This rationale comes from the fact that a neural network with a surrogate loss function on return allows for a more direct and nuance understanding of the complex nonlinear relationships between signal scoring and returns. Traditional linear models have some difficulty to capture these complexities, where a neural network can adjust dynamically to a broader range of data patterns. The use of a surrogate loss especially enables the model to focus on long-term returns and portfolio performance rather than just fitting the data.

As a result, the signal-weighted solutions should likely outperform the naïve diversification of 1/n, but the neural network approach in particular should outperform the naïve 1/N baseline method as it is better suited to capture underlying patterns in the data and adjust weights accordingly.

**Section 5: Results**

In order to evaluate the effectiveness of the proposed method, we implemented and tested several portfolio optimizations techniques using our own code from scratch. All optimization logic (including signal-based scoring, signal-weight learning, and MVO solvers) was developed from scratch, in accordance with the learning objectives of the course. Our primary goal was to test whether the signal-weighted optimization techniques could consistently beat the naïve 1/N benchmark allocation strategy.

**5.1 Dataset and Experimental Setup**

The experiment here was ran on a pool of 20 companies from the S&P 500 index, using daily historical stock data spanning from 2011 to 2020. The data was sourced from the Yahoo Finance dataset, which contains adjusted closing prices used to compute returns and signal scores. We computed scores at every trading date for three momentum-based signals: Simple Moving Average (SMA), Relative Strength Index (RSI) and Moving Average Convergence/Divergence (MACD). These scores were used as input for both the signal combination and optimization stages.

The data was organized to support yearly training and evaluation using a rolling window approach. Each model was trained on 12 months of historical data (based on annualized returns) and then evaluated on the 12 months that followed a one-year gap, ensuring a full-year buffer between training and evaluation to prevent data leakage in the returns. Performance was assessed using both the annualized returns and the cumulative returns as evaluation metrics.

**5.2 Methods Compared**

We evaluated the performance on multiple different approaches using a simple linear model:

1. Signal Optimization (no maximum threshold):

The model learns a weighted combination of signals to directly maximize the portfolio return by selecting the single best-scoring stock, without considering a variance-based risk measure.

1. MVO with Signals (no maximum threshold constraint, with volatility measure):

The model learns a weighted combination of signals and uses Mean-Variance Optimization (MVO) to select the best-scoring stock from the pool, incorporating both signal scores and portfolio variance.

1. MVO with Signals (with 20% Max Constraint):

Similar to the method above but introduces a 20% maximum allocation constraint per asset. Hence, the optimizer selects the top 10 stocks in our pool, ensuring better diversification and more realistic, investable allocations.

1. Neural Network with Softmax Surrogate Loss:

The model learns signal weights based on a neural network approach that uses a softmax surrogate loss to directly optimize portfolio returns. This aims to better capture the complex interactions between the signals and optimize allocations in a more flexible and data-driven way.

**5.3 Result Summary**

|  |  |  |
| --- | --- | --- |
| Method | Average difference per year in Annualized Return | Avg diff / year total return |
| Signal Optimization (no maximum threshold) | -0.0015 | -0.3652 |
| MVO with Signals (no maximum threshold constraint, with volatility measure) | 0.0429 | 1.20 |
| MVO with Signals (with 20% Max Constraint) | 0.0404 | 1.13 |
| Neural Network with softmax surrogate loss | **0.4189** | **0.4734** |

**5.4 Analysis**

From the results, we can see that the Neural Network with Softmax Surrogate Loss outperforms all the other methods by a substantial margin. The neural network model achieved a significant improvement in return, with an average annualized return of 0.4189 and a total return difference of 0.4734. This suggests that the neural network, which uses a surrogate softmax loss function, is better suited capture the complex relationships between the signals and the returns compared to the simpler linear models.

However, it is also important to note that the MVO-based optimization methods also performed better than the 1/N benchmark by a slight margin with annualized returns of 0.0429 and 0.0404. These methods benefited from the incorporation of both signal scores and volatility measures.

On the other hand, the Signal Optimization withtout volatility measures did slightly worse than the 1/N benchmark, with a negative annualized return of -0.0015. This result shows that the linear approach with no risk management did not adequately capture the dynamics of the stock market and led to a slight suboptimal performance.

The introduction of the Neural Network with Softmax Surrogate Loss significantly improves upon the previous linear methods by learning a more flexible and adaptive model, which can handle the non-linear relationships between the signals and the portfolio returns. These results also demonstrate the potential of using deep learning models in financial portfolio optimization to improve decision-making to better manage risk-return trade-off and warrant further exploration in future research.

**5.5 Notes on Discrepancies**

While these results demonstrate a significant improvement by using a neural network approach, it is still crucial to note that the performance of models in financial markets is highly dependent on the quality of the input signals, the market conditions and the model tuning. As such, further robust exploration is required before making a definitive claim about its effectiveness.

However, it can be reasonable to assume that if the neural network indeed generalizes better in most scenarios, it would be due to its ability to adapt to the underlying patterns in the data, while also being more sensitive to hyperparameter choices and the quality of its training data.

Moreover, the discrepancy in performance between the linear-based methods and the neural network approach highlights the added value of using more complex, data-driven models in portfolio optimization. Although these models may not be as light or easy to implement, their ability to process intricate patterns in data offers an edge in performance that is significant. Therefore, finding the right balance of complexity and performance is crucial in achieving optimal results.

Hence, while the linear models focus on balancing returns and risk using a variance-based method, the neural network model optimizes with a more dynamic learning process that allows for a better, broader capture of market signals and adapts to market changes more effectively.

**Section 6. Self-Evaluation**

**6.1 Comment and Assessment**

Problem Description

I believe that I carefully framed a meaningful research problem that is both relevant to the course but also aligned to the current trends at the intersection of machine learning and operational research in the real-world application of systematic finance.

Methodology

The methodology that was developed was rigorous, innovative, and logically consistent with the other sections of the report. This end-to-end architecture that learns to optimize portfolio signal weighting is an idea that pushes forward the integration of machine learning and optimization techniques in the domain, where this type of research is still emerging. I believe the difficulty is quite advanced, as the project required building and validating a full pipeline from signal processing to optimization from scratch in a relatively short time frame.

Discussion

The discussion section successfully connects the project to the broader topics that are covered in the course. I feel that the analysis was thoughtful and showed how integrating learning and optimization can address this real-world context and its challenges.

Question and Hypothesis

I believe that the question was relevant, practical, and intellectually stimulating as it requires deep reflection and critical thinking. I believe that it is a complicated question that, if solved fully, could bring meaningful insight into the investment field. The hypothesis follows this question and has a good intuition to it, that is consistent with the theoretical foundations discussed in the course and in the literature.

Results

The results are consistent with the experiments that were conducted. They are aligned with the expected trends based on the theoretical intuition, demonstrating that the implementation and reasoning behind the experiments were sound.

**6.2 Reflection**

I am particularly happy with the originality of the project and the level of ambition. Instead of reproducing a basic approach, I was allowed to explore an advanced methodology that involved significant technical challenges, including the design of full end-to-end full machine learning and optimization financial architecture for finance from scratch. This allowed me to learn a great deal about all the fields connected to this project. I am also very satisfied with how clearly the report explains the project and its challenges, with well-structured sections that connect in a coherent and logical way to form a strong narrative.

If I had more time or could start over, I would define a consistent methodology for generating and evaluating the results earlier in the project. This would have allowed me to compare my experiments earlier in the project and draw conclusions faster. That said, I still believe that I made the right choices given the knowledge and constraints at the time and adapted well throughout the project as I learned new things.

Overall, I am proud of the technical depth, creativity, and clear integration with the course topic and will push this project further after this class.

**Section 7: References**

The knowledge base for this project was primarily derived from the material that was presented in the course. Especially, the material surrounding SLO, ILO and SPO. Here are a few of the references used:

* Sadana, A., Feng, S., & Wilder, B. (2025). *A Survey of Contextual Optimization Methods for Decision-Making Under Uncertainty.* European Journal of Operational Research.
* Loke, Y., Tang, Z., & Xiao, C. (2022). *Decision-Driven Regularization: A Blended Model for Predict-then-Optimize.* Advances in Neural Information Processing Systems (NeurIPS).

AI Tools Usage

I used AI tools such as ChatGPT to assist in drafting, structure sections of the report, correcting some spelling mistakes, reformulating sentences to better illustrate my points, and providing guidance during the coding and debugging process.