Sharpe Ratio Optimization

Stefan Knapik (stefanknapik99@gmail.com)
Renee Zhuang (rzhuang22@students.d125.org)

Mentors: Chanwook Park (chanwookpark2024@u.northwestern.edu)
Prof. Wing Kam Liu (w-liu@Northwestern.edu)
Prof. Mark Fleming (Mark.Fleming@Northwestern.edu)
August 11, 2021
Mechanistic Data Science Summer 2021

Northwestern | McCORMICK SCHOOL OF ENGINEERING

Introduction

- Maximize **profit** & minimize exposure to **risk**
- Stock prediction is almost **impossible**
- **Portfolio**: combination of investments held by an individual
- Portfolio diversification
  - Strategy that **minimizes risk** by mixing a wide variety of investments
  - “not putting all the eggs in one basket”

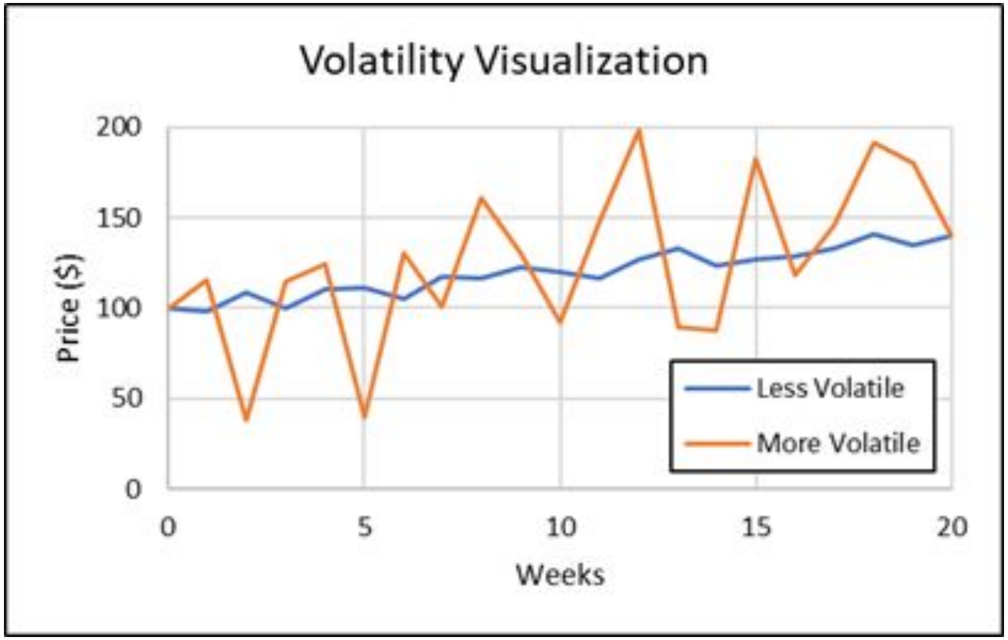
Introduction

Return is a profit from an investment

Volatility

- Measure of the variation in returns with time
- Volatile portfolios are considered riskier

The two illustrated assets have the same average return, but the returns of the more volatile orange line have a larger standard deviation than the blue.



Introduction

- Difficulties of portfolio distribution
  - Financial markets change ➞ assets negatively correlated could become positively correlated
  - Huge number of available assets

We consider directly optimizing the return per unit risk of a portfolio by maximizing its Sharpe ratio.

⬇

$$\boxed{\text{Sharpe ratio} = \frac{\text{mean return} - \text{risk free return}}{\text{standard deviation of returns}}}$$

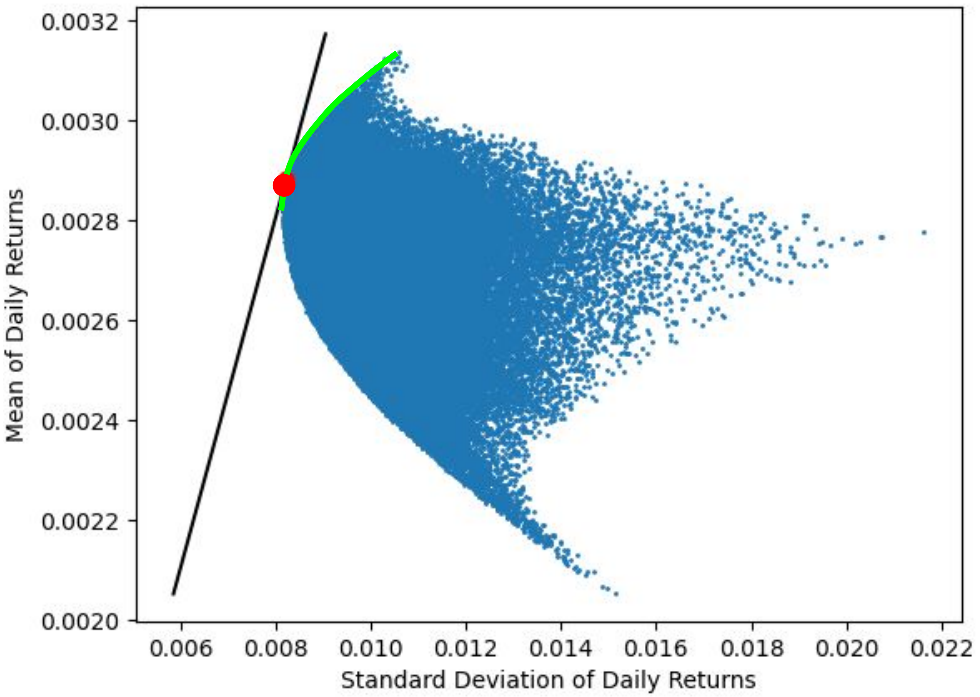
Background

Harry Markowitz's "Modern Portfolio Theory" is a mathematical framework for assembling a portfolio to maximize return per risk

- Has practical limitations and makes unrealistic assumptions

Monte Carlo simulation is popular

- e.g. randomly generating 100,000 portfolios and picking the best one
- 4 assets were considered for the plot to the right
- Efficient Frontier in green



Background

Even Monte Carlo simulation has its own limitations in accuracy and the number of assets it can consider for a portfolio

It's like randomly throwing darts, hoping one lands near the "best spot"

- darts are cheap, but not free
- as the size of the dartboard increases, it becomes increasingly less likely that a finite sample of darts hits near the "best spot"

Each row is made up of binomial coefficients - bolded row is C(n,100)

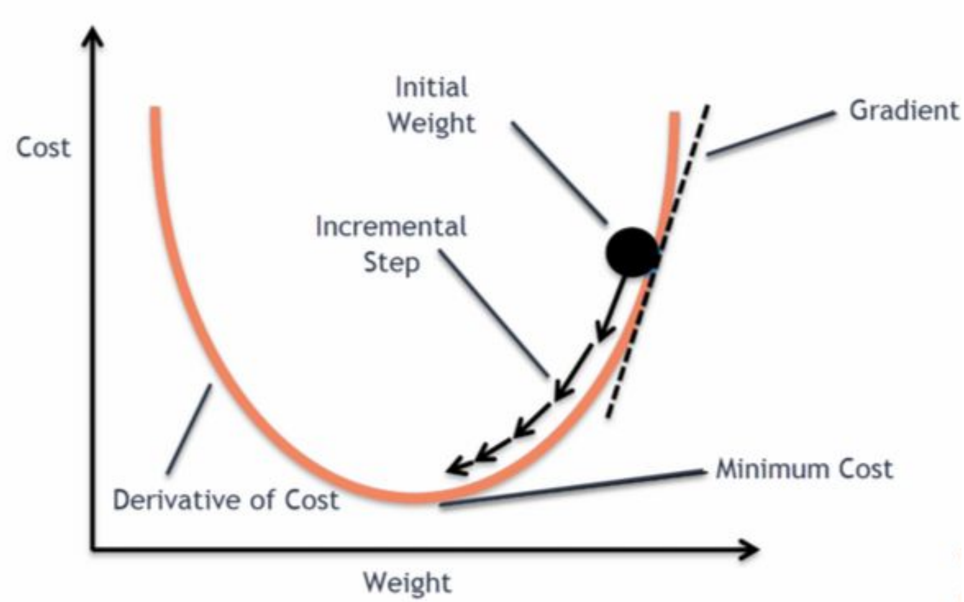
Number of Possible Portfolios

		Number of Assets									
		3	4	5	6	7	8	9	10	20	50
Portfolio Segment Size	10%	66	286	1001	3003	8008	19448	43758	92378	2.0E+07	6.3E+10
	5%	231	1771	10626	53130	230230	888030	3.1E+06	1.0E+07	6.9E+10	1.2E+17
	2%	1326	23426	316251	3.5E+06	3.2E+07	2.6E+08	1.9E+09	1.3E+10	4.6E+16	5.0E+28
	1%	**5151**	**176851**	**4.6E+06**	**9.7E+07**	**1.7E+09**	**2.6E+10**	**3.5E+11**	**4.3E+12**	**4.9E+21**	**6.7E+39**

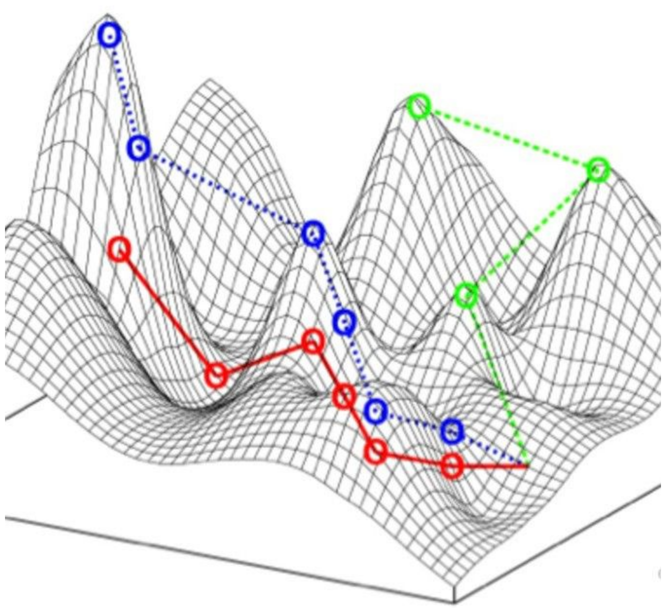
Approach

Gradient ascent makes it possible to iteratively converge on the portfolio with the maximum Sharpe ratio

- Taking steps with the independent variables in the direction of desired change for the dependent variable
- Only requires that our objective function be differentiable
- Should allow us to consider much more than 4 assets



https://blog.clairvoyantsoft.com/the-ascent-of-gradient-descent-23356390836f



https://medium.com/gradientascent/trending

Northwestern | ENGINEERING

System and Design

The user of our model will **input a list of potential assets** they are considering investing in.

⬇

Our model will **return the optimal distribution** of these assets in a portfolio using historical price data.

System and Design

Eliminate the “risk free return” constant from the Sharpe ratio in our objective function ➞ maximum of objective function and maximum of original Sharpe ratio will be reached by the same portfolio distribution.

$$\boxed{\text{Sharpe ratio} = \frac{\text{mean return} - \cancel{\text{risk free return}}}{\text{standard deviation of returns}}}$$

$$\boxed{\text{Objective function} = \frac{\text{mean of returns}}{\text{standard deviation of returns}}}$$

For a given portfolio distribution and lookback window, we can calculate the net return of the portfolio for each day to help us compute the mean and standard deviation in our objective function.

Data Collection

Stock data is readily available online through Yahoo and Google Finance

For use in Python, we will use the IEX finance API

```
from iexfinance.stocks import Stock
tickers = Stock(['TSLA','DIS','AAPL','MSFT','VTI', 'BND','DBC','VOO'])
tickers.get_advanced_stats()
```

	beta	totalCash	currentDebt	revenue	grossProfit	totalRevenue	EBITDA
TSLA	2.203425	17141000000	14877000000	35940000000	7611000000	35940000000	4594000000
DIS	0.868889	15890000000	26642000000	58383000000	17880000000	58383000000	7301000000
AAPL	1.582528	69834000000	106385000000	325406000000	129776000000	325406000000	99820000000
MSFT	1.313473	125407000000	72193000000	159969000000	109389000000	159969000000	76074000000
VTI	1.012068	None	None	None	None	None	None
BND	0.029802	None	None	None	None	None	None
DBC	0.384036	None	None	None	None	None	None
VOO	0.998968	None	None	None	None	None	None

Feature Selection

We are assuming that recent historical price trends and volatilities are indicative of current market conditions

Below is the structure of our input matrix

$$\begin{array}{rc}
 & \begin{array}{ccccccc} \multicolumn{2}{c}{\text{stock 1}} & \multicolumn{2}{c}{\text{stock 2}} & \cdots & \multicolumn{2}{c}{\text{stock n}} \end{array} \\
\begin{array}{r} \text{day 1} \\ \vdots \\ \text{day 100} \end{array} & \left[\begin{array}{cc|cc|c|cc} p_{1,1} & r_{1,1} & p_{2,1} & r_{2,1} & & p_{n,1} & r_{n,1} \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ p_{1,100} & r_{1,100} & p_{2,100} & r_{2,100} & & p_{n,100} & r_{n,100} \end{array}\right]
\end{array}$$

$\mathbf{p_{i,t}}$ is the close price of asset i for day t

$\mathbf{r_{i,t}}$ is the return of asset i for day t

Gradient Ascent for Optimization

Recall the Sharpe ratio $\approx \frac{\text{mean}(R_p)}{\text{standard deviation}(R_p)}$

We can maximize the objective function $L_T = \frac{\text{mean}(R_{p,t})}{\sqrt{\text{mean}(R_{p,t}^2) - (\text{mean}(R_{p,t}))^2}}$

$\frac{\partial L_T}{\partial \gamma}$ is readily calculable via backpropagation

$\gamma_{new} = \gamma_{old} + \alpha \frac{\partial L_T}{\partial \gamma}$ where α is the learning rate for gradient ascent

Iterate until we converge on the maximum L_T

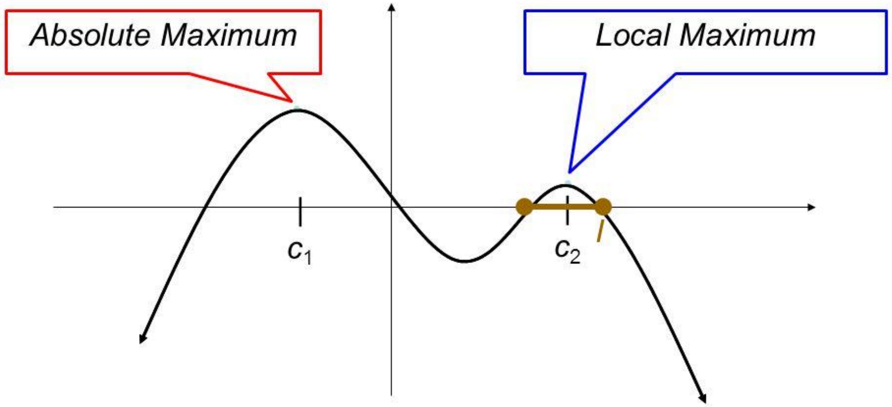
Gradient Ascent for Optimization

Internal parameters: weights and biases in the neural network

Internal parameters (γ) work like dials which we can adjust to change the output of the neural network.

By seeing which way we need to turn each dial and making incremental adjustments, we can arrive at a maximized Sharpe ratio.

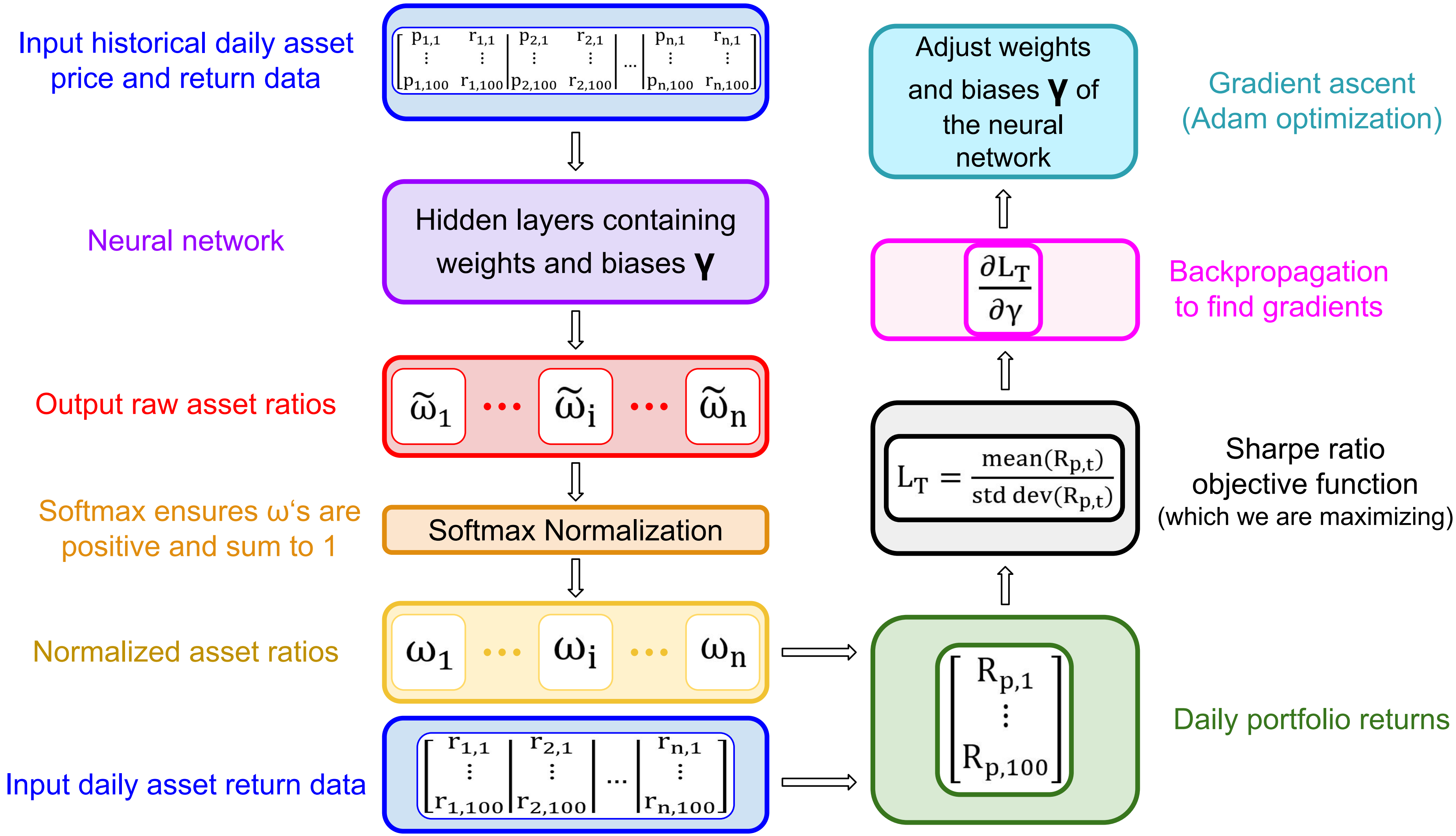
- We used **Adam optimization** to avoid settling into local maxima



https://www.tekportal.net/absolute-maximum/

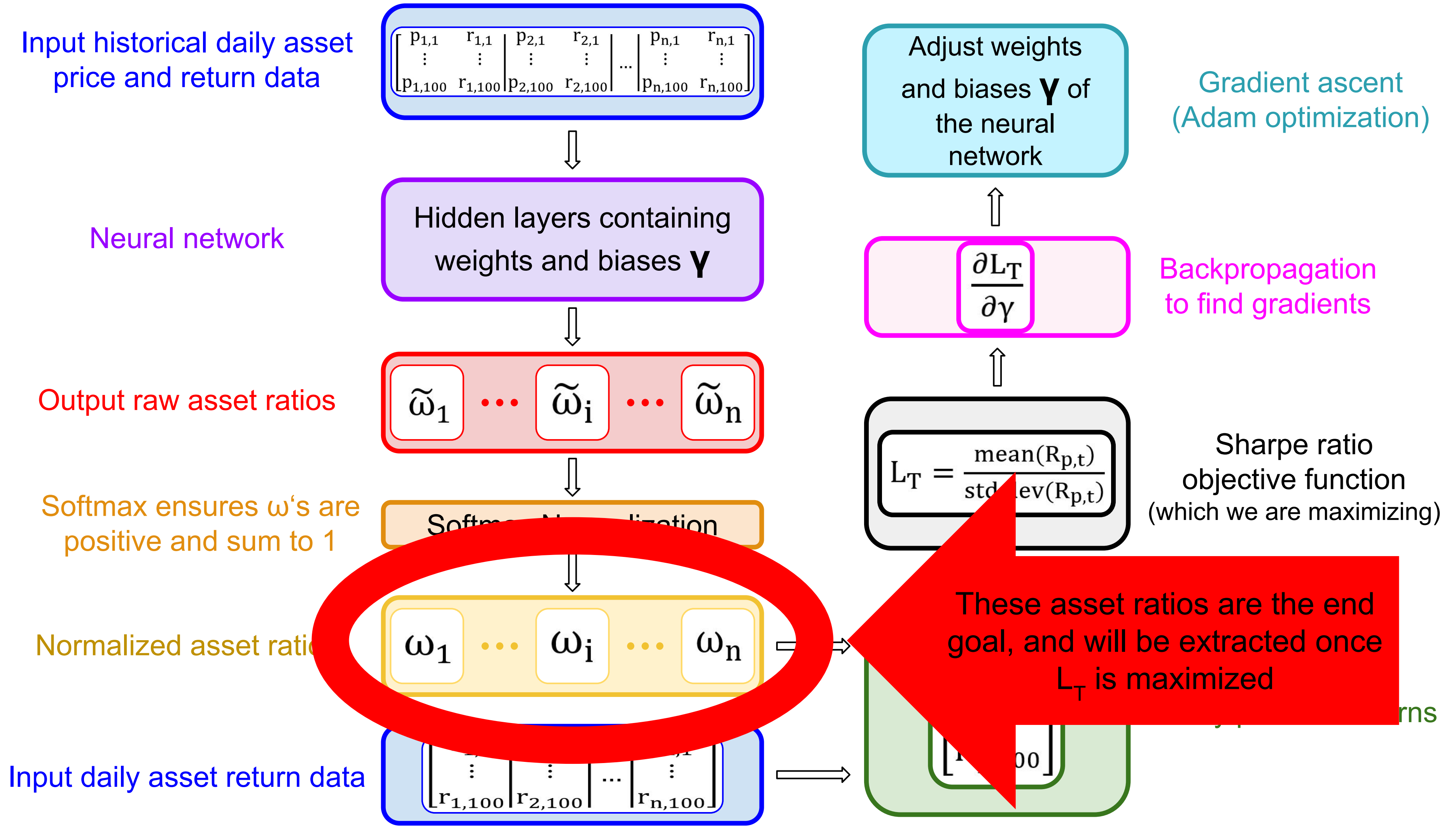
Model Architecture

For a portfolio considering n assets, we will repeat to converge on the maximum L_T



Model Architecture

For a portfolio considering n assets, we will repeat to converge on the maximum L_T

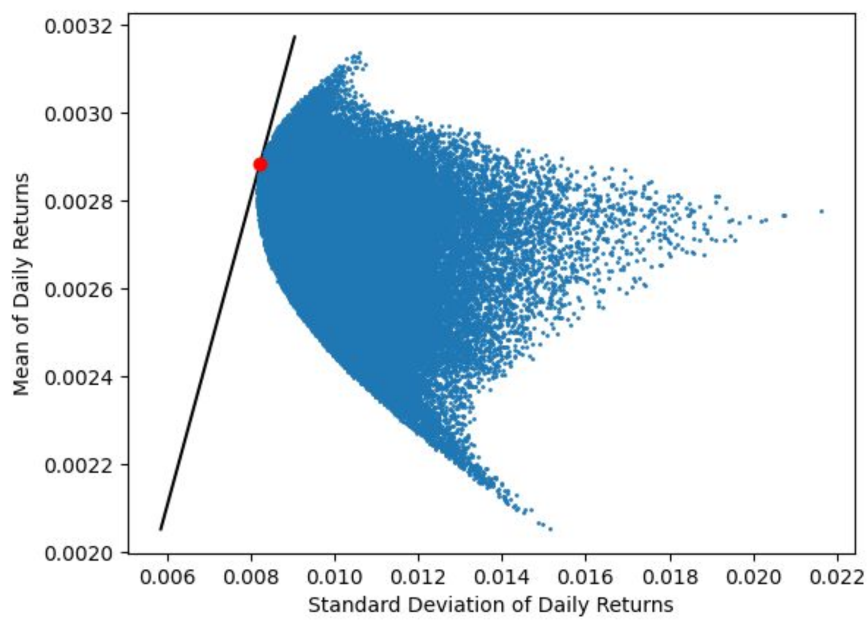


System and Design

Once the Sharpe ratio is maximized, we can use the normalized asset ratios (from the output layers of the neural network) to construct our ideal portfolio.

Normalized asset ratios $\omega_1 \cdots \omega_i \cdots \omega_n$

Most returns per risk in past 100 days → expected to perform similarly well in the near future



Hidden Layer Architecture

Right now:

- Feed Forward Neural Network (FFNN)
- 1 hidden layer
- 64 neurons
- ReLU activation function

This simple architecture performed as well as deeper ones with more hidden layers for optimization on a single lookback window.

For the future:

- Train the model with data from many lookback windows
- Recurrent Neural Network (RNN)
- Long Short Term Memory (LSTM)

These continuations will turning the scope of this problem from simple optimization (only considering the data we have) into machine learning (generalizing the model for data we don't have yet).

Results of Gradient Ascent

Here we performed 100 iterations of gradient ascent on the price data for Apple, Tesla, Microsoft, and Google from the past 50 days.

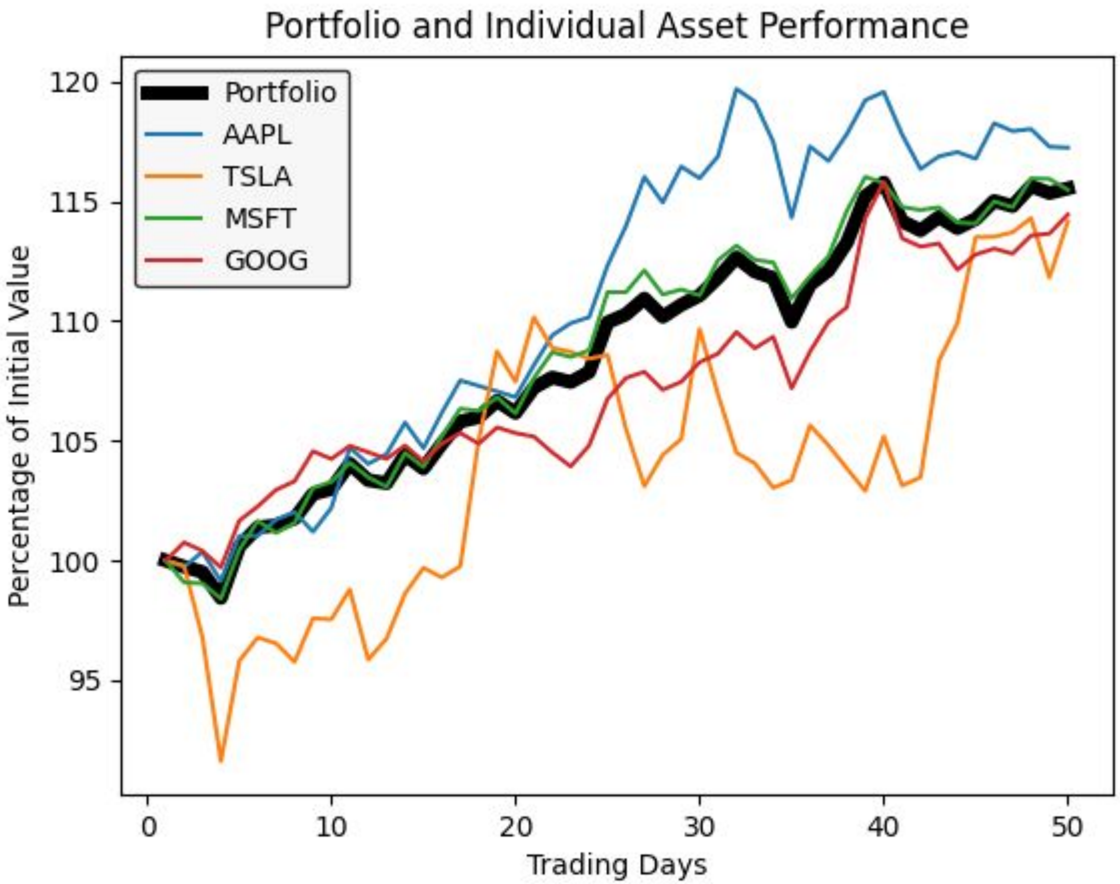
```
  1  Sharpe:    0.325        Asset Ratios:  0.251  0.248  0.251  0.251
  2  Sharpe:    0.328        Asset Ratios:  0.254  0.240  0.253  0.253
  3  Sharpe:    0.332        Asset Ratios:  0.258  0.230  0.256  0.256
  4  Sharpe:    0.336        Asset Ratios:  0.263  0.216  0.261  0.260
  5  Sharpe:    0.341        Asset Ratios:  0.269  0.200  0.266  0.265
 20  Sharpe:    0.360        Asset Ratios:  0.298  0.052  0.337  0.312
 40  Sharpe:    0.363        Asset Ratios:  0.220  0.068  0.384  0.328
 60  Sharpe:    0.363        Asset Ratios:  0.167  0.070  0.467  0.296
 80  Sharpe:    0.363        Asset Ratios:  0.179  0.071  0.459  0.291
100  Sharpe:    0.363        Asset Ratios:  0.177  0.070  0.457  0.295
```

Our Sharpe objective function (not precisely the Sharpe ratio) increases with each step as the 4 asset ratios are updated.

Performance Over the Lookback Window

This shows how a portfolio constructed from the final realized asset ratios would have performed during these past 50 days.

Asset Tickers:	Ideal Ratios:
AAPL	0.177
TSLA	0.070
MSFT	0.459
GOOG	0.294

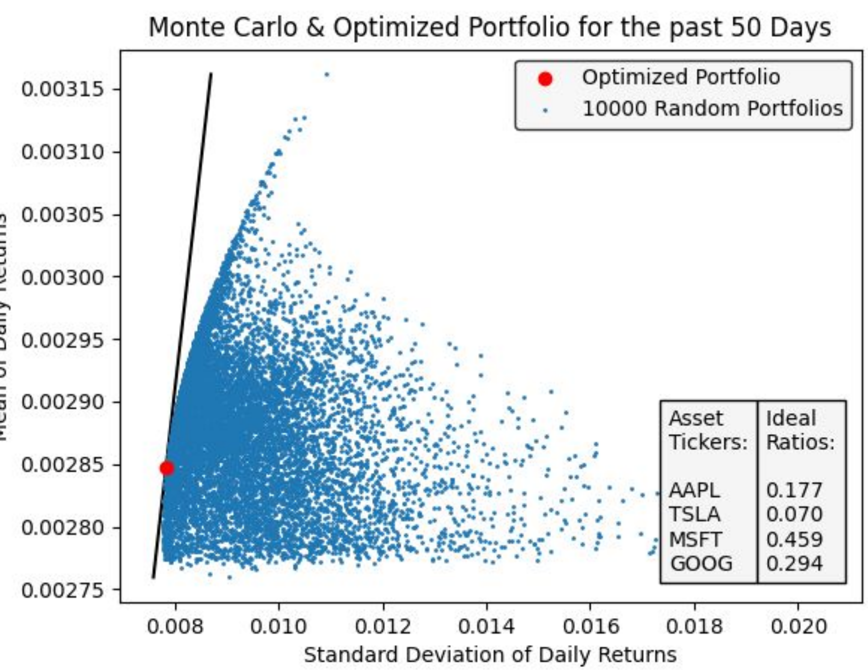


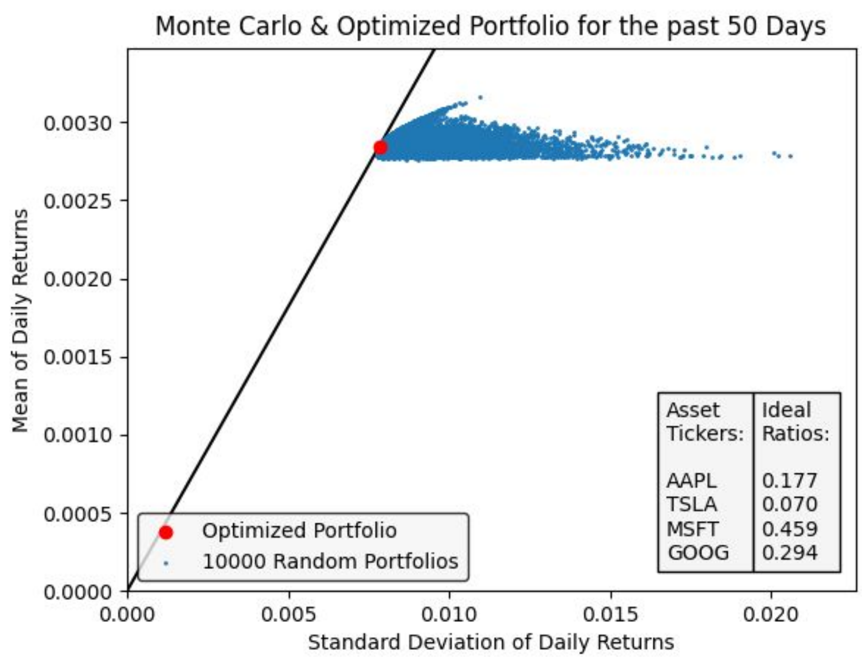
Validation with Monte Carlo Simulation

For these 4 tech stock, using the past 50 days of data, we can compare our optimized portfolio to the results of Monte Carlo simulation.

10k random portfolios are enough to illustrate the efficient frontier.

The optimized portfolio truly has the highest Sharpe Ratio.



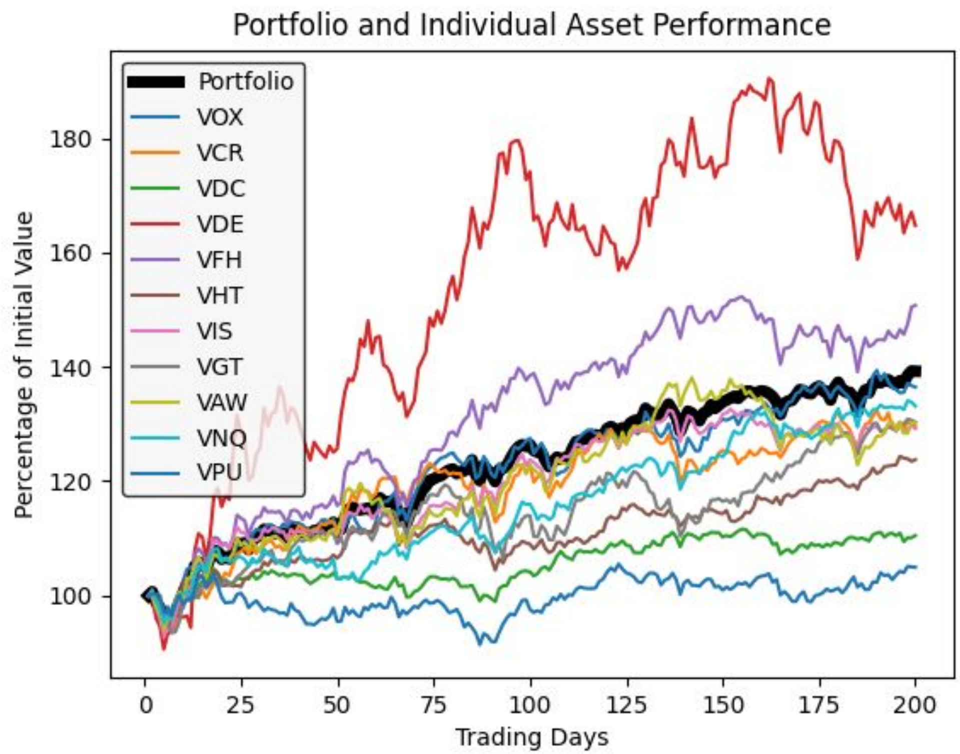


11 Vanguard sector ETFs

There are 11 sectors in the stock market, each with different industries.

We found Vanguard ETFs that tracks each sector and performed our model on the past 200 trading days of data.

Asset Tickers:	Ideal Ratios:
VOX	0.320
VCR	0.000
VDC	0.000
VDE	0.000
VFH	0.364
VHT	0.172
VIS	0.000
VGT	0.000
VAW	0.000
VNQ	0.144
VPU	0.000



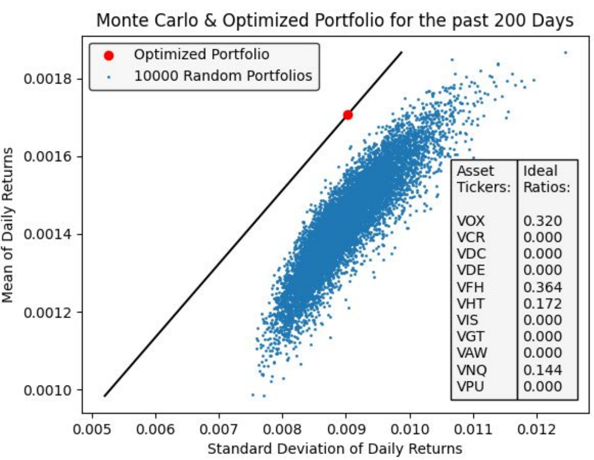
Comparison to Monte Carlo Simulation

Now with 11 possible assets, the space of possible portfolios is too large to cover by randomly guessing. More assets makes it even worse.

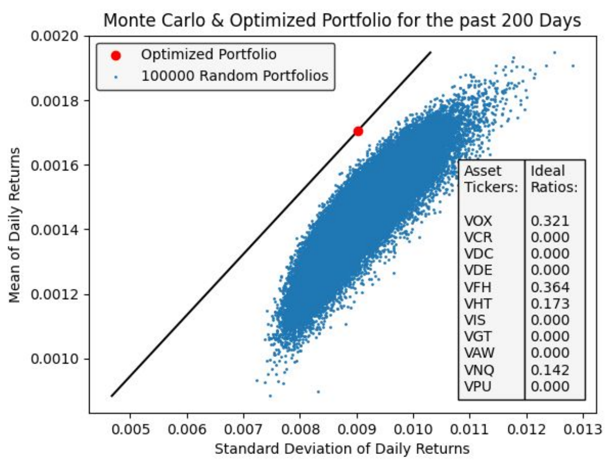
Out of 1 million random portfolios, none are close to the optimized one.

- 1 million guesses took 10 minutes to perform on a laptop
- the optimization took 6 seconds

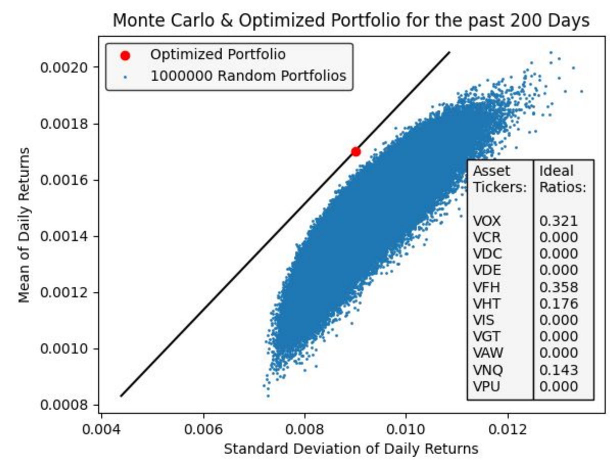
10k portfolios



100k portfolios



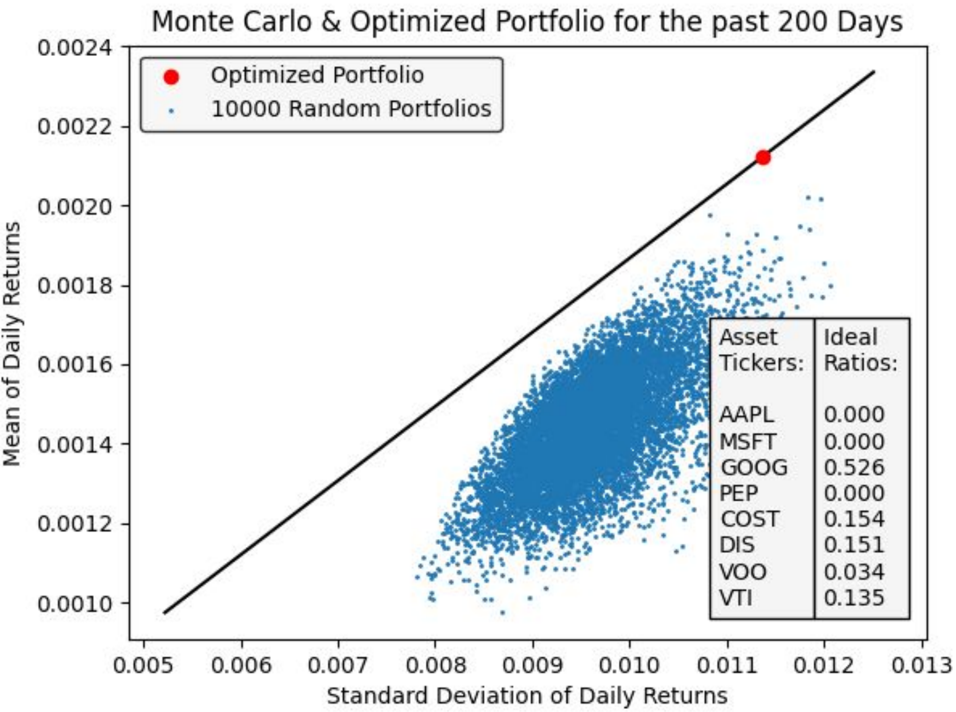
1 million portfolios

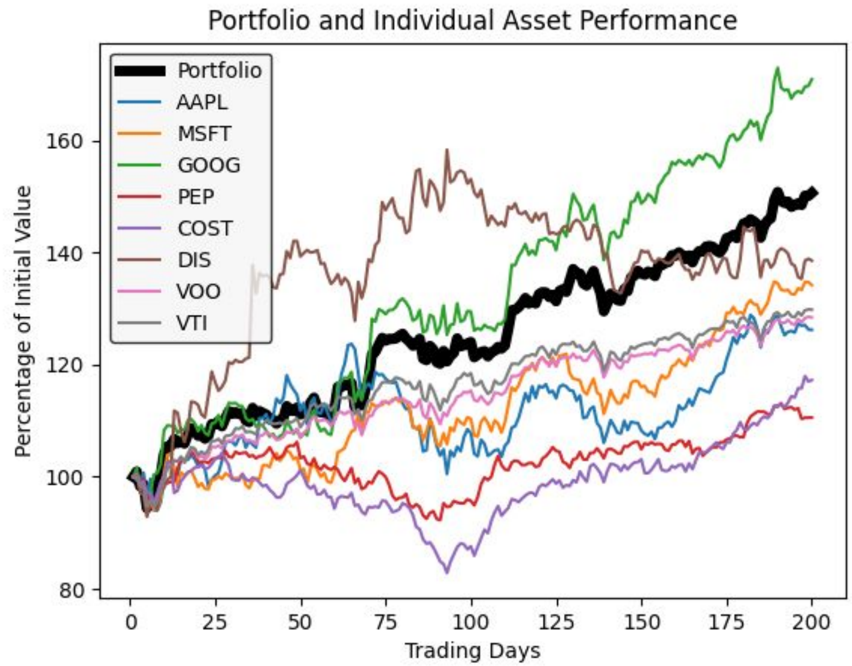


Mixed Stocks and ETFs

Prices and returns are features common across many types of assets, which allow us to create hybrid portfolios constructed from nearly any type of security.

Here the lookback window is 200 days





Limitations

Sharpe ratio uses standard deviation as a proxy of risk

- assumes that returns are normally distributed

The presented algorithm is optimization, not learning yet

- we only use one lookback window, and we optimize the portfolio for performance over that single period of time
- there is no generalization to the data we don't have yet
- our model is only useful for portfolio rebalancing under the assumption that recent past performance indicates near future performance

Future Investigation

Train the model using batches of many lookback windows pulled from a larger dataset and iterate for a certain number of epochs

Develop code for backtesting

- We can use historical data to simulate an investor rebalancing their portfolio every week, month, or year using this model

Hidden layers with recurrent architectures

- Simple RNN
- LSTM
- Gated RNN

Other objective functions

- Sortino ratio - negative deviation is the proxy for risk
- Modified sharpe ratio - recent information weighted heavier
- Portfolio diversification degree
- Any other differentiable function!