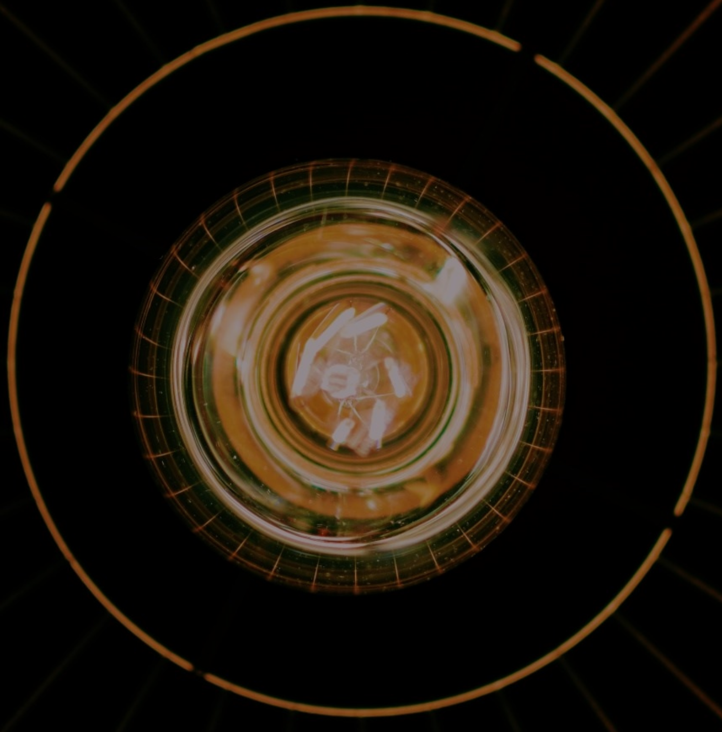
New Model of AI Driven Asset Management
Higher Alpha with Lower Cost through AI Technologies

QRAFT AI Quant Series

QRAFT Deep Learning Asset Allocation

Manager, Hanwook Jeong
Manager, Chanmin Jun
Senior Manager, Eunchong Kim

August 2020



Contents

QRAFT Technologies
+82 2 487 8555
qraft@qraftec.com

[Home page] https://www.qraftec.com/
[Subscribe to Newsletter] https://www.qraftec.com/newsletter

Copyright 2020. QRAFT Technologies All rights reserved

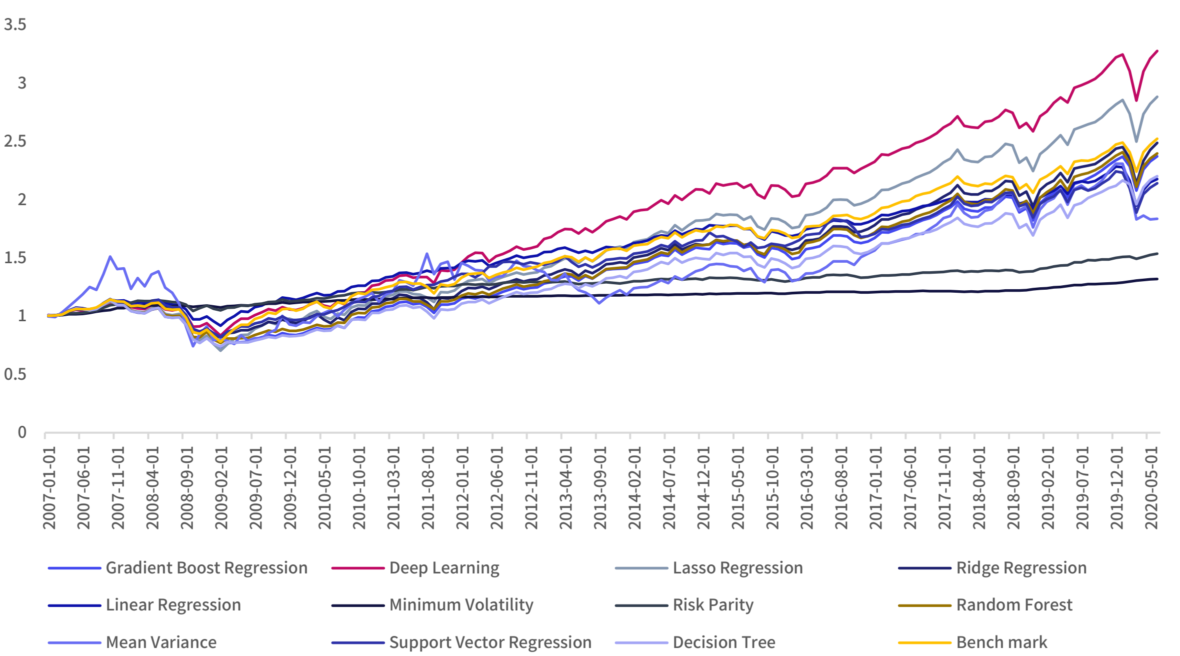
Summary

Asset allocation refers to the portfolio construction strategy by appropriately allocating proportions to available asset classes. With asset allocation, investors can significantly reduce the overall risk of portfolio by investing in assets ranging from high risk assets such as stocks, to assets with low risks such as bonds and cash. Increasing the rate of return is crucial for an investment, but reducing risk is equally important to achieve a steady return. Our purpose of asset allocation using deep learning is to provide the model that delivers better returns with similar risks compared to existing well-known benchmarks.

Traditional asset allocation investment is widely adapted by many asset managers in various forms, and individuals also utilizes these strategy without difficulty. The traditional asset allocation method assumes the continuity of the historical distribution that the past data distribution will be similar in the future, so it is hard to consider to be realistic with our view in the market. In addition, even if tactical asset allocation such as Momentum strategy and Long-Term Reversal strategy for portfolio performance is implemented, we cannot be sure whether specific parameters are optimal. To solve these problems and to build better strategies, we use artificial intelligence technology rather than human trial-and-error. With A.I. driven process, it is possible to minimize the time spent to optimize well-known quant strategies and deliver better risk-adjusted return. When examining the performance of asset allocation through deep learning with simple empirical analysis, it showed better performance than the not only traditional asset allocation strategy, the 60/40 portfolio and the risk-parity strategy but Machine Learning Methodologies.

We effectively establish investment strategy and execute process using financial data through A.I Technologies, even achieve better performance of strategies. The Asset Allocation strategy utilizing Deep Learning is expected to lead to a chance in the financial market, as there is probability for simplifying the process and improving performance as the technology develops.

Portfolio Performance



Source: QRAFT Technologies, Compustat, DataStream, FRED

Introduction

Asset allocation is one of the elements of a successful investment strategy. The assets available for investment have become more diverse in recent years. Asset allocation means that the portfolio is constructed by appropriately allocating the investable asset classes. With asset allocation, investors can significantly reduce the overall risk of portfolio by investing in assets ranging from high risk assets such as stocks, to assets with low risks such as bonds and cash. Increasing the rate of return is crucial for an investment, but reducing risk is equally important to achieve a steady return. The goal of our asset allocation is to provide an asset allocation model which provides better returns with the same level of risks compared to existing well-known benchmarks. Therefore, it is also necessary to focus on reducing risks through asset allocation. When approached simply from a return perspective, too much weights are allocated to specific asset classes, making it difficult to adequately diversify risk.

1. Traditional Asset Allocation

60/40 portfolio is one of the well-known traditional asset allocation strategies. This constitutes of 60% stocks and 40% bonds portfolio and is known for being more robust in terms of risk compared to investing in just one asset. In particular, the above portfolio can be easily recommended to individual investors since the allocation method is easy while the benefit of diversification is huge.

Markowitz (1952)[1] pioneered modern portfolio theory, arguing that the portfolio should consider the risk-return relationship between individual assets and the entire portfolio as well as the risk-return of each asset. Through the past return and covariance matrix of a given asset class, investors can construct the portfolio maximum expected return within a specific risk. As shown in the first equation below, we can use the incremental increase in marginal risk and rate of return to find weight vector with maximum Sharpe ratio. Also, we can also build Minimum-Volatility portfolio by finding the weight that minimizes the variance of the entire portfolio as shown in the second equation below.

$$\text{Mean-Variance Portfolio} : \underset{w}{arg\ max}\ \frac{\mu' w - r_f}{\sqrt{w'\Sigma w}}$$

$$\text{Minimum-Volatility Portfolio} : \underset{w}{arg\ min}\ w'\Sigma w$$

μ : Expected Return

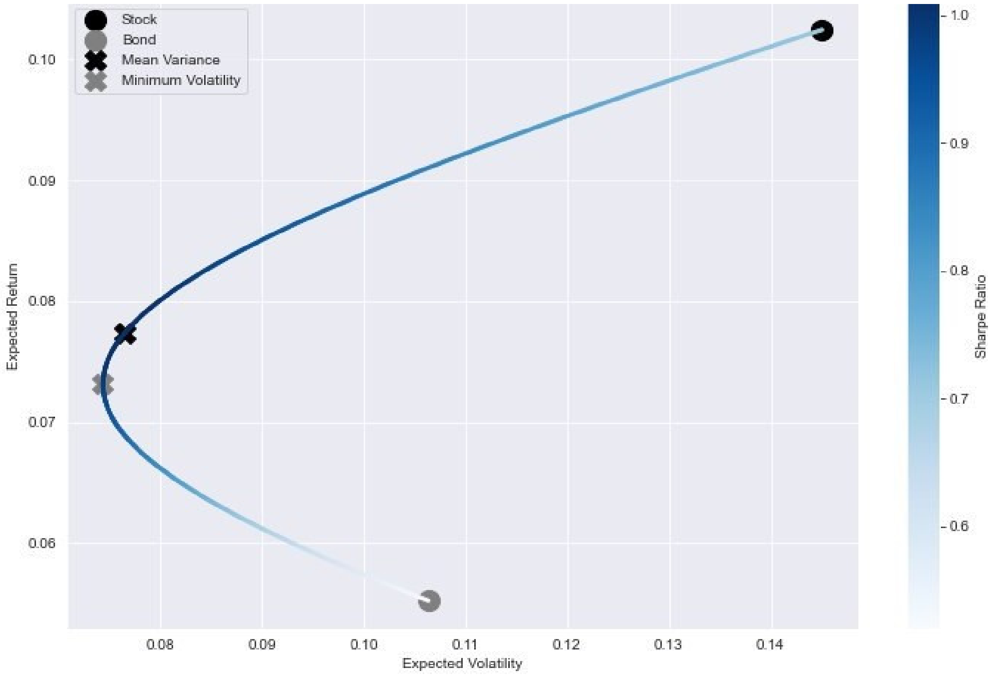
w : Individual Asset Weight

Σ : Covariance Matrix of Asset Returns

r_f : Risk-free Rate

[1] Harry Markowitz, 1952, Portfolio Selection, The Journal of Finance 7-1, 77-91

[Figure 1] Risk & Returns Relationship



Source : QRAFT Technologies, Compustat

A risk parity is an asset allocation strategy focusing on portfolio risk. The strategy equalizes the risk contribution of each asset class in the portfolio. The risk level of the asset classes is equally weighted, and the impact on the overall portfolio performance is same for each asset class. The risk of a given portfolio can be expressed as the sum product of the risk contribution and the weight of the individual assets.

$$\text{Risk Contribution of Individual Assets : } \sigma_i(w) = w_i \times \frac{\partial\sigma}{\partial w_i} = w_i \times \frac{(\Sigma w)_i}{\sqrt{w'\sigma w}}$$

$$\text{Marginal Risk Contribution of Individual Assets : } \frac{\partial\sigma}{\partial w_i} = \frac{\sigma_{i,P}}{\sigma} = \rho_{i,P} \times \sigma_i$$

$$\text{Portfolio Risk : } \sigma_P = \sum_{i=1}^{N} w_i \times \frac{\partial\sigma}{\partial w_i}$$

w: Individual Asset Weight

σ : Standard deviation of portfolio return

ρ : correlation coefficient

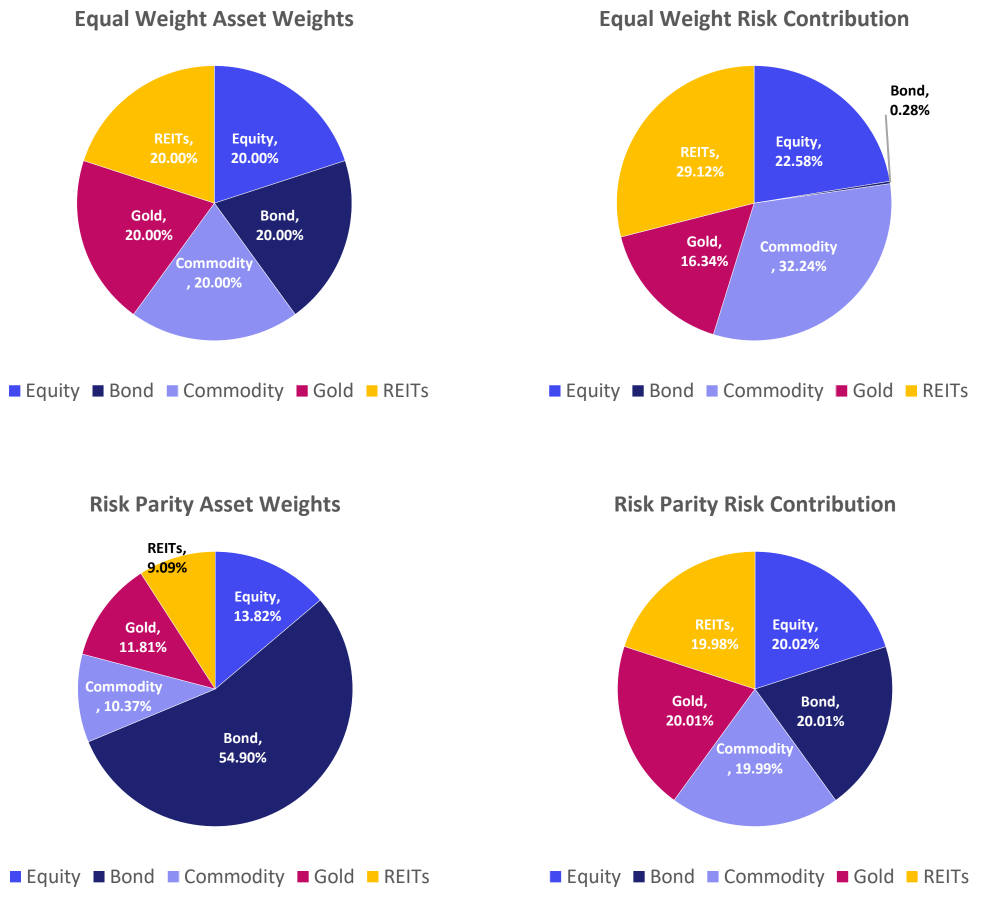
i: Individual Asset

P: Portfolio

The algorithm finds a weight vector that the risks that individual assets contribute are the same to the whole portfolio. We set the objective function of the following equation and derive the weights of assets by optimization algorithm.

$$\underset{w}{arg\ min} \sum_{i=1}^{N} \left[w_i - \frac{\sigma(w)^2}{(\Sigma w)_{i,N}} \right]$$

[Figure 2] Risk Exposure of Equal Weight Portfolio & Risk Parity Portfolio



Source : QRAFT Technologies, DataStream

The above [Figure 2] shows the result of equal weight portfolio and risk parity portfolio. Investable assets consists of stocks, bonds, commodities, gold and REITs. First, in the case of equal weight portfolio, stock comprises 20% of total portfolio, while the risk it contributes to the portfolio is 22.58%. When examining the risk parity portfolio, we can see that all five asset classes contributes equal risk to the portfolio. In other words, it is hard to argue that risk allocation is optimal since most of the volatility comes from risky assets, not the safe assets such as bonds. The risk parity strategy can significantly reduce downside by equally allocating the source of the risks.

The All-Weather Portfolio[2] is also one of the risk parity strategies, which emerged through criticism of the existing asset allocation methods. BridgeWater Associates argue that most institutional portfolios lack diversification which leads to suboptimal returns when adjusted to assumed risk of the portfolio (Bridgewater Associates, 2009) . In other words, most investors are fully aware of the benefits of diversification but focus more on the allocation of capital rather than risk. However, the existing risk parity strategy is based on historical correlation and volatility and is very sensitive to economic conditions. This is because the correlation between asset classes is not constant and the risk itself is difficult to predict. To overcome this limitation, the All-Weather strategy takes into account the structural relationships of asset classes in different economic environments and minimizes the impact of unexpected economic changes.

[2] BridgeWater, 2009, The All Weather Strategy

[Figure 1] Benefit of different asset classes on different macro environments

	Growth	Inflation
Rising	**25% of Risk** - Equity - Corporate Spread - Commodities - EM Debt Spreads	**25% of Risk** - Inflation-Linked Bonds - Commodities - EM Debt Spreads
Falling	**25% of Risk** - Nominal Bond - Inflation-Linked Bonds	**25% of Risk** - Nominal Bonds - Equities

Source : QRAFT Technologies

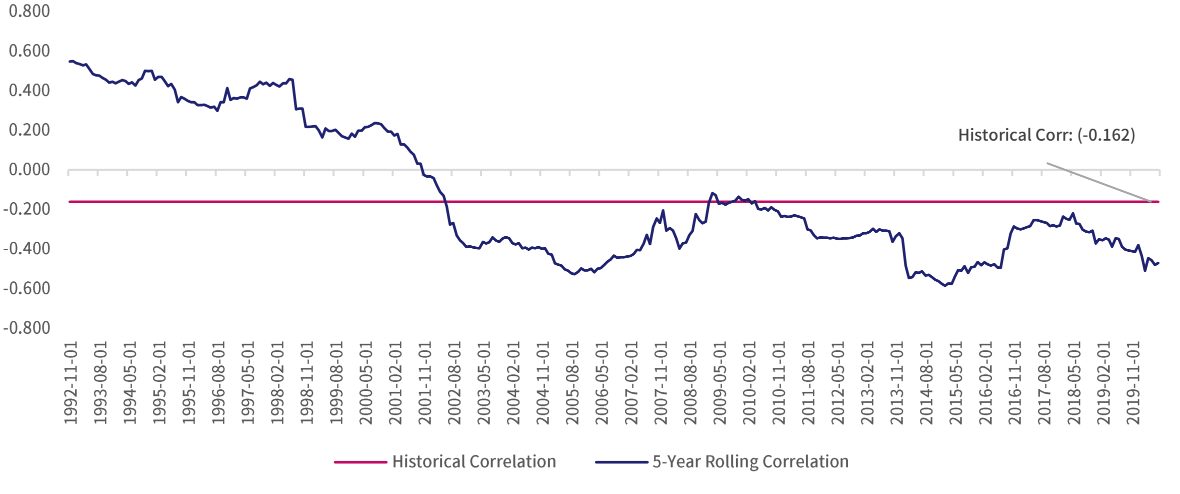
[Table 1] above shows how different macroeconomic environments affect each asset class in the All Weather strategy. By balancing risk in four market environments, the portfolio's sensitivity to either environment can be minimized to consistently derive the asset group's risk premium, and the portfolio can be more robust than the traditional asset allocation and risk parity.

2. Why Deep Learning?

Traditional asset allocation investment is used by many asset managers in different forms, and individuals can also easily utilize these methods. However, this traditional portfolio construction method does not imply a predictive model. The traditional asset allocation method assumes the continuity of the historical distribution that the past data distribution will be similar in the future, so it is hard to consider realistic with our view in the market.

For example, we consider the correlation between stocks and bonds has a negative(-) correlation in general, but the relationship between the two assets has undergone dynamic changes as shown in [Figure 3] below. In order to maximize the effect of asset allocation, it is necessary to construct a portfolio appropriately to respond to different dynamics that change with time. This is partly in line with All Weather mentioned earlier in that it takes a strategy to adapt to changing market conditions.

[Figure 3] S&P 500 and U.S Treasury Note's 5-Year Rolling Correlation

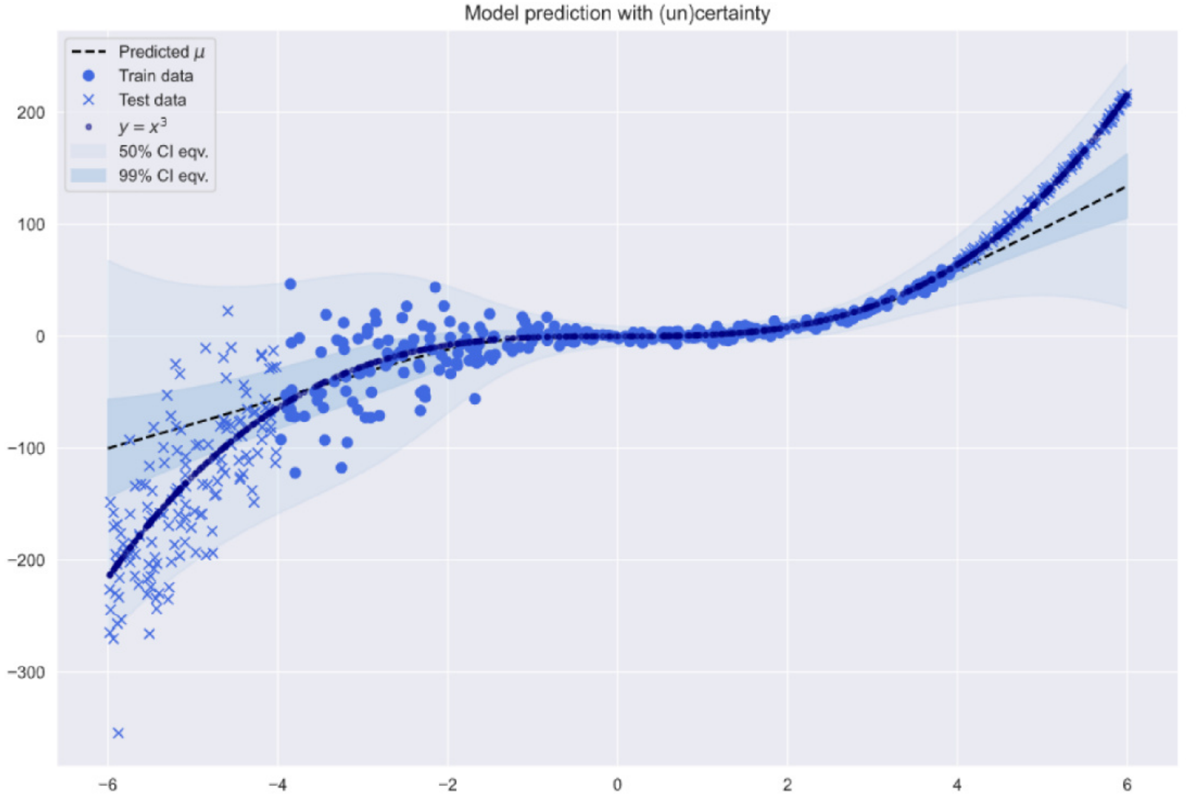


Source: QRAFT Technologies, Compustat

The use of quantitative strategies such as momentum and long-term reversal for portfolio performance is called tactical asset allocation. However, we encounter problems for selecting appropriate metrics, for example, the optimal period for measuring the momentum value, and the criteria for judging the long-term reversal. These criteria are often determined by prior studies, and many resources must be allocated to test each case. To solve these problems and to achieve better models, we use artificial intelligence techniques rather than human trial-and-error. With A.I. driven process, it is possible to minimize the time spent to optimize well-known quant strategies and deliver better risk-adjusted return. It can also efficiently allocate investment assets and flexibly respond to dynamic market conditions.

According to Olalekan (2016)[3], deep learning models can capture nonlinear relationships. When applying statistical time series modeling such as autocorrelation or moving average, the order of the model must reflect how much the window should be for the time series data. Several statistical techniques are used to estimate the order of the model, but most statistical time series models based on linear models have limitations in dealing with data with very complex nonlinear relationships.

[Figure 4] Model Prediction with (un)certainty



Source : QRAFT Technologies

[Figure 4] above is an example of a data prediction of nonlinear relationship through our deep learning model. There are nonlinear relationships among many financial data, and it can be inappropriate when linear estimation is made. Therefore, when using financial data, deep learning model can have better predictive power when used properly.

[3] Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans, 2016, Nonlinear System Identification Using Deep Dynamic Neural Networks

3. Primary-Mission for Utilizing Deep Learning

Neural Network (NN) which is the basic structure of deep learning, has one of many advantages in flexible structure design. This means that deep learning models can be appropriately used to address the limitations of historical data analysis. In addition, while traditional machine learning methodologies are used to specific tasks, NN can freely configure the structure and inner workings of the model, so that many different tasks can be conducted such as regression, classification, dimension reduction, distribution estimation, and data generation (sampling). This flexibility is used in all aspects of strategy design and it helps reduce the resources needed in the overall process. In the learning process, continuous learning is performed with the recent data, and factor formation is optimized based on features extracted from both the cross-section macro data and asset group data. However, there are some complications that need to be addressed in advance when applying deep learning models. There are three main types of problems that we discuss in this research: insufficient sample data, data noise, and overfitting problems.

(1) Sample Data Deficit

The current asset allocation model aims to derive an optimized portfolio from learning labelled data. Two prerequisites are required for deep learning, namely that "significant" and "sufficient" data must exist. The 'significance' means how to calculate factors such as value, momentum, and regression to the mean at the asset group level. To derive this significance, we train deep learning models to calculate momentum values. Next, we face the question of how much data is enough. Sufficient sample data is required to improve the predictive power of the model, but there are only too short time series compared to the available features. For example, for 40-year data, only 480 samples are available for monthly basis. In this case, you may face problems such as The Curse of Dimensionality. Asset allocation requires long-term time series data. We consider using data from at least 1980 should be used in the training set, and it is necessary to consider long-term economic cycles, short-term economic cycles, and short-term factor phenomena simultaneously. For non-existent data, various measures such as MICE imputation or NaN Embedded Layer[4] is used. This paper describes the results with MICE Imputation.

(2) Data Noise

$$AR(1)\colon P_t = \alpha P_{t-1} + \varepsilon_t$$
$$P_t = P_{t-1} + I_{t-1} + \varepsilon_{t-1}$$
$$P_t - P_{t-1} = I_{t-1} + \varepsilon_{t-1}$$
$$P_t : Asset\ Price_t$$
$$I_t : Information_t$$
$$\varepsilon_t : Noise_t$$

Second, there is the noise problem of the data. There is noise component in most of the time series itself. When the stock price follows the AR(1) model, the next day's stocks price is determined by previous stock price + information + noise. What the deep learning model needs to capture here is information. However, the noise component is usually larger than the information component. The information component cannot be reliably measured, and as a result, the best predicted value for the next time period becomes the current value. Therefore, it is important to measure information by effectively removing the noise. To effectively remove noise contained by financial time series, we use moving average (MA, EMA, etc.) and bilateral filter, which are used commonly for the purpose (Sylvain Paris, 2009[5]).

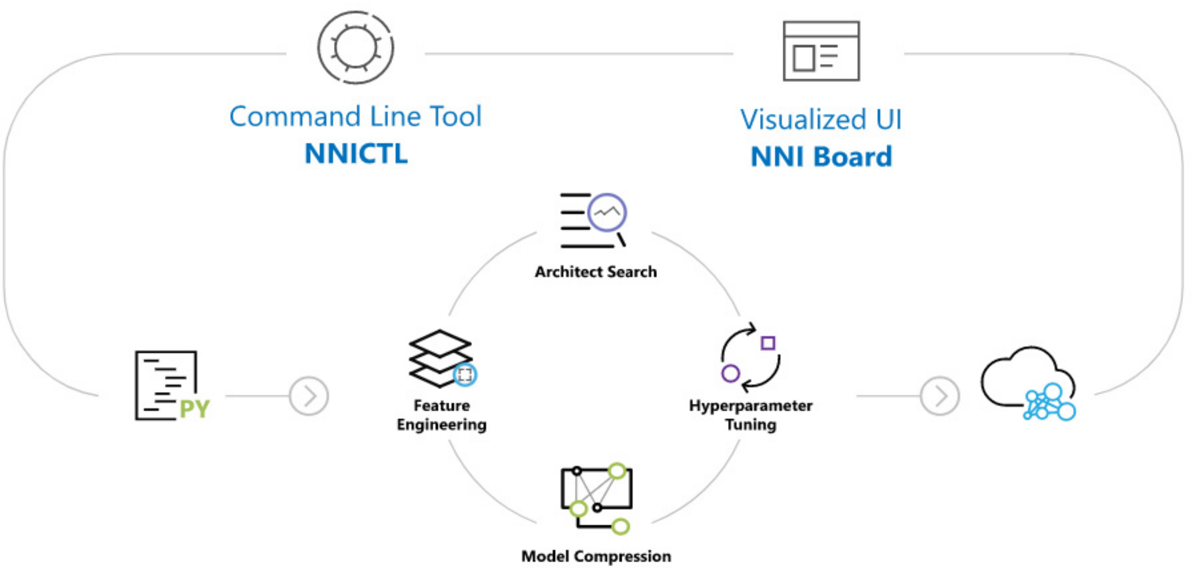
[4] It is a Neural Network developed independently by QRAFT Technologies, effectively masking Not Available Values that are prevalent in financial data, and auto-scaling only necessary information to map with embedded features. Through this method, missing data can be easily and effectively processed without forward-fill or masking.
[5] Sylvain Paris, Pierre Kornprobst, Jack Tumblin and Fredo Durand, 2008, Bilateral Filtering : Theory and Application, Foundations and Trends in Computer Graphics and Vision 4-1, 1-73

In addition, ADF-Test is used to verify the stationarity of the data. Without the stationary data, there is a high possibility of deriving a false correlation in the time series. These traditional noise reduction methods improve learning for deep neural networks compared to data without processing. However, input data must be pre-processed prior to the learning without the forward information. Therefore, the denoising module based on CNN Stacked Auto Encoder[6] is also used to automatically remove the noise from the input data during the learning process.

(3) Over-Fitting

Lastly, there is the possibility of Over-Fitting. In machine learning, the number of hyperparameters is large, and the difference between the in-sample performance and the out-of-sample performance of the model can significantly diverge depending on how the hyperparameters are set. Therefore, minimizing the hyperparameters of the network and optimizing the network structure helps prevent Over-Fitting. To this end, we use Auto-ML modules such as NNI from Microsoft to search predetermined hyperparameters state space and automate the optimal structure construction.

[Figure 5] NNI Flow



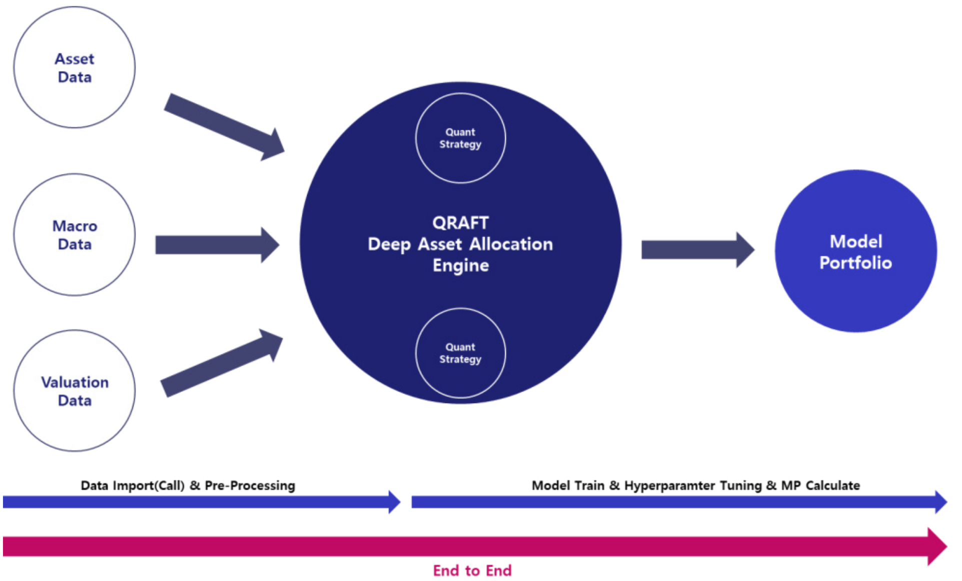
Source : Microsoft

When using the return data, the probability of Over-Fitting is very high because the model learns noise inherent in the data. Therefore, our asset allocation model does not use direct return prediction model, such as traditional stock price or asset group return prediction. After calculating the optimal portfolio separately and numerically, deep learning model trains to the weight label and indirectly learn the behavior. By adopting the behavior rather than prediction movements of assets, the possibility of overfitting can be greatly reduced, and the deep learning model can be better utilized for investment uses.

[6] CNN(Convolutional Neural Network) based Auto Encoder algorithm. Stacked Auto Encoder is auto encoder with many hidden layers. Stacked auto encoder has symmetric structure with centered hidden layers, The auto encoder encodes original data with noise and decodes the original data with noise. When trained, the encoder can be used as a denoising algorithm.

QRAFT Deep Learning Model Structure

[Figure 6] AI Engine Procedure



Source: QRAFT Technologies

[Figure 6] shows core features of our deep learning model. Our model calls data, preprocess data in the format that the model learns, tunes hyperparameters, and finally, constructs optimal model portfolio. The entire process from end to end is done all at once. It helps when given the problem definition and data for asset allocation, it is relatively easy to obtain any desired model portfolio within the investable universe. The following paragraphs explain each step described above.

1. Data Set

There are three types of data used for the model: asset index data to measure actual portfolio composition and performance, macroeconomics data for regime detection, and valuation data. Our proprietary Kirin API helps using these data sources and utilizing them with high flexibility.

(1) Asset Class Index : For the model DataStream and Compustat data are used. Instead of processing individual security data of each asset, model training uses representative index data. It is because the length of the index data is long, and ETFs benchmarking those indices are available for investment.

(2) Macro Economics Data : We use FRED data through Kirin API, calling Credit Spread, Inflation, WTI Oil Price, GDP, M1, M2, Effective Federal Funds Rate, etc.,

(3) Valuation Data: Using Compustat via Kirin, we load data that represents the current market valuation such as S&P 500 P/E and dividend yield.

The three types of data go through deep learning optimization process with a pre-labeled optimal portfolio as the target value. Through this process, it aims to generate higher risk-adjusted return by constructing an optimized model portfolio.

2. Pre-Processing

In order to address issues such as lack of sufficient sample and stationarity of the data as described above, MA denoising, stationarity test through ADF and data imputation are performed.

(1) Denoise & Stationarity Check

[Figure 7] MA Denoise

Source : QRAFT Technologies, FRED

[Figure 7] Above is an example though the simplest MA Denoise method among noise removal methods. The amount of information is estimated by removing the noise of difference in time series data.

[Figure 8] Stationary Check

```
=====================Check Stationarity (ADFuller test)==========================
WIL5TMK p-value : 7.48202131907021e-06
MSEMKF$ p-value : 4.0239756236835386e-05
SPBDUSL p-value : 1.7071725646858624e-06
SPBDU10 p-value : 5.028928383007938e-05
SPBDUS3 p-value : 0.04432610979079157
GSCITOT p-value : 0.00021297831399425123
GSGCTOT p-value : 9.77953408972412e-05
BAAFF p-value : 0.001962927039840011
PCETRIM12M159SFRBDAL p-value : 0.0002042919647095844
GDPC1 p-value : 0.0006391248511992836
WTISPLC p-value : 6.411045104513693e-05
M1 p-value : 0.3436136929891016 <-- may not be stationary
M2 p-value : 0.19682106061738136 <-- may not be stationary
T10Y2Y p-value : 0.0018071585730344054
UNRATE p-value : 0.174699662372714 <-- may not be stationary
REAL_VOL p-value : 0.0027749940871733838
S&PPE p-value : 8.20997196839016e-05
S&PDIV p-value : 3.8921349975745376e-06
DFF p-value : 1.519000249968902e-05
=================================================================================
```

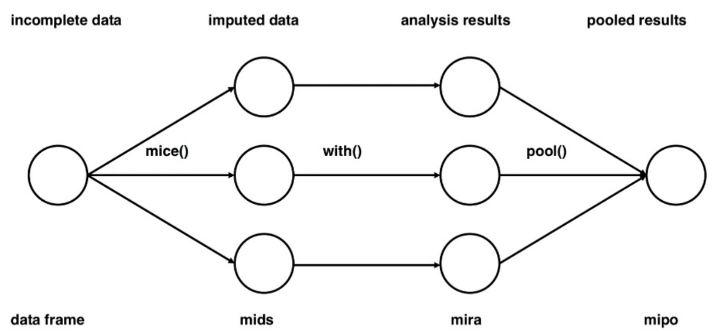
Source: QRAFT Technologies

Our pre-processing module automatically performs unit-root test through ADF test as shown in [Figure 8] and maintains the stationarity of input data by performing pre-processing, as necessary. In order to check the feasibility of the model, it is essential to verify the stationarity of the input data. Through this process, it is possible to greatly reduce the possibility of overfitting to a specific period in the past due to the characteristics machine learning.

(2) Imputation

Financial data used often has missing values due to various reasons, such as randomly missing or the limitation of collection period. This imposes many limitations on data analysis, and we utilize inferring likely data using relationship present in currently available data.

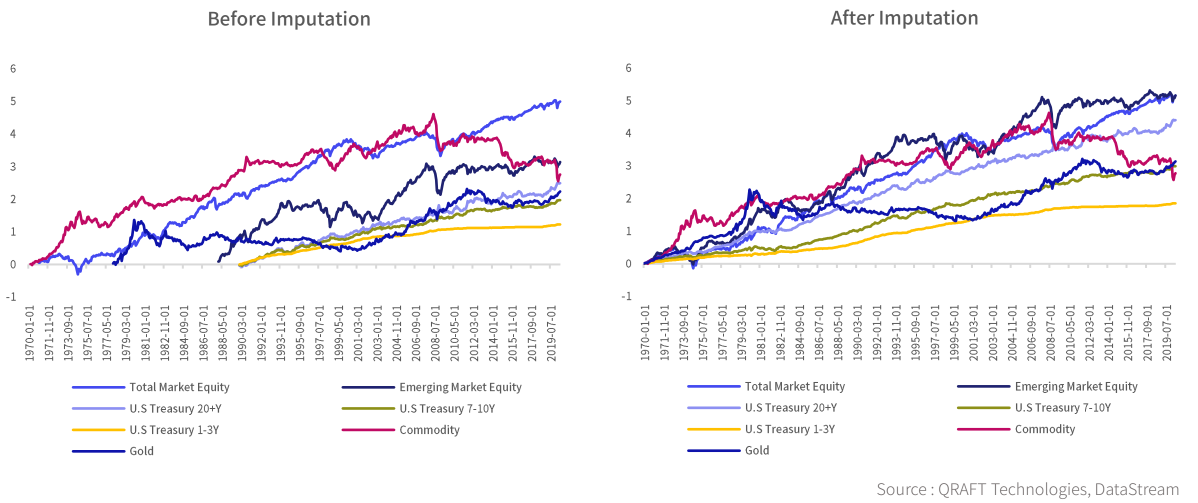
[Figure 9] Main Steps used in Multiple Imputation



Source : MICE : Multiple Imputation by Chained Equations in R

[Figure 9] shows a schematics of Multiple Imputation by Chained Equation (MICE). Past missing data is estimated via MICE imputation method used for substitute data. MICE can be used not only for time series data, but also for data missing due to industrial classification, and adjustment for outliers. [Figure 10] below is an example of data imputation using MICE.

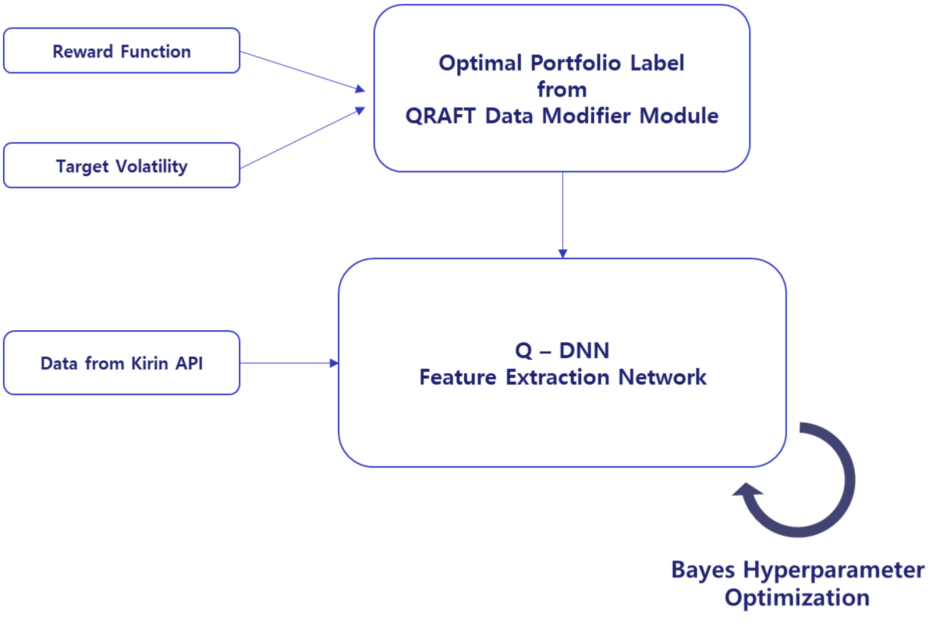
[Figure 10] Before & After Imputation



Source : QRAFT Technologies, DataStream

3. Model Structure

[Figure 11] Structure of Model



Source : QRAFT Technologies

[Figure 11] shows the schematics of the deep learning engine used by Qraft Technologies. To solve any real-life problem using deep learning, three things are needed. The first is to define what you want to learn, the data to make model learn its objective functions, and the model that is effective in structure and hyperparameters that learns the problem well.

When applying deep learning to asset allocation, simply trying to predict returns or predict movements may not be appropriate. Individual asset classes, such as stocks, bonds, and commodities, not only have very different characteristics, but also affected by different macroeconomic indicators among different regimes, and their interaction on the entire portfolio may not be as expected. Therefore, in order to reflect various factors and interactions indirectly in our model, we aim not to predict the returns of asset classes, rather to learn and predict the weight of an optimal portfolio. In other words, taking the optimal action with the present data is learned indirectly by mimicking the constructed optimal portfolio under the assumption that we can accurately know the future. To this end, we define a portfolio performance metric (reward) that can reflect the investor's objectives and restrict the target volatility that reflects the investor's risk tolerance level. Additionally, a discount rate that can reflect the uncertainty of future behavior is applied when labelling the optimal portfolio weight.

After generating the optimal behavioral label, it is necessary to construct a model structure that can extract the characteristics of the data well. In order to construct the optimal portfolio, we use our Kirin API to collect macroeconomic data and valuation data reflecting the specific time period. Look-a-head and other issues are addressed internally in Kirin API, and the deep learning model can learn optimal behavior using only the data that were available at each point in time.

It is hard to describe in full specificity for the deep learning model used for many products, but the key components are as follows. To learn the complex nonlinear relationship from the given financial data, all the recurrent type, fully connected and convolutional neural networks are used where necessary. Recurrent networks are proved efficient in learning time series, while fully connected and convolutional networks are known for dealing with more static data. Specifically, we use our own TS-CS module[7] designed for financial time series, which can extract both the cross-sectional and time-series features simultaneously. Adversarial training can be additionally implemented to extract features robust from noise.

4. Hyper Parameter Tuning

Hyperparameters are the variables responsible for generating model structure. The examples include learning rate, mini batch size, L2 Regularization rate. For hyperparameter tuning for each deep learning models, some priorities have been internally discussed as follows:

(1st) Network Capacity: Capacity is defined by components such as the number of nodes and layers of neural network model. Insufficient capacity leads to underfitting because the model is not able to learn necessary features, while too large capacity leads to overfitting.

(2nd) Learning Rate & Batch Size: Capacity is defined by components such as the number of nodes and layers of neural network model. Insufficient capacity leads to underfitting because the model is not able to learn necessary features, while too large capacity leads to overfitting by memorizing features specific to training dataset.

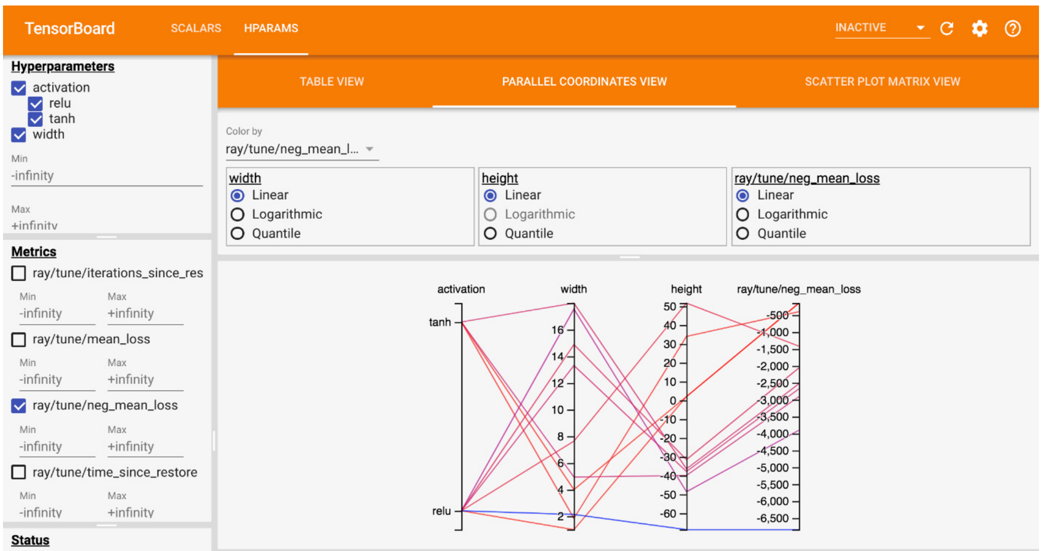
(3rd) Target Portfolio Hyper Parameters: It is necessary to obtain well-labelled targets for the portfolio. Hyperparameters to be considered when configuring the optimal portfolio include lambda value, window size of the label portfolio, and objective function (Sharpe Ratio, Sortino Ratio, etc.).

(4th) Early Stopping: When the model runs too many iterations, it leads to overfitted result, while too few iterations cause underfitting. Early stopping aims to prevent either case by stopping its optimization based on score function.

[7] Neural network structure developed by Qraft Technologies. The network effectively extract features cross sectional and time series simultaneously

Hyperparameters tuning process mainly involves two stages. The first step is to directly check the learning curve with a Tensor Board, etc. while experimenting manually. For example, if the batch size increases, adjust the appropriate learning rate, and observe if short-term underfitting occurs in the above-mentioned early stopping process. The second step is to create an Auto-ML environment then using specific modules such as NNI or Ray Tune. The performance of the model varies greatly depending on how well the model is tuned. It is important to make good use of the Ray Tune and Auto-ML algorithms because it is practically impossible to tune every time.

[Figure 12] Hyperparameter Tuning with Ray Tune



Source : Tune : Scalable Hyperparameter Tuning, docs.ray.io

Application

With a simple example we compare the differences in portfolios. To check the excellence of the model, we will look at the methods that are frequently used for asset allocation. The traditional 60/40 portfolio is benchmarked, and the risk parity portfolio (RP), average-variance portfolio (Min-Vol) are reviewed. Next, along with our deep learning model (DL), we will look at the portfolio results by applying various machine learning (ML) techniques by changing only the model to the same learning environment. We aim to check the superiority of our DL method compared to the existing ML method. ML method for comparison in this paper uses Support Vector Regression (hereinafter SVR), Decision Tree (DT), Linear Regression (LR), Gradient Boosting Regression (GBR), Random Forest (RF), Ridge Regression and Lasso Regression.

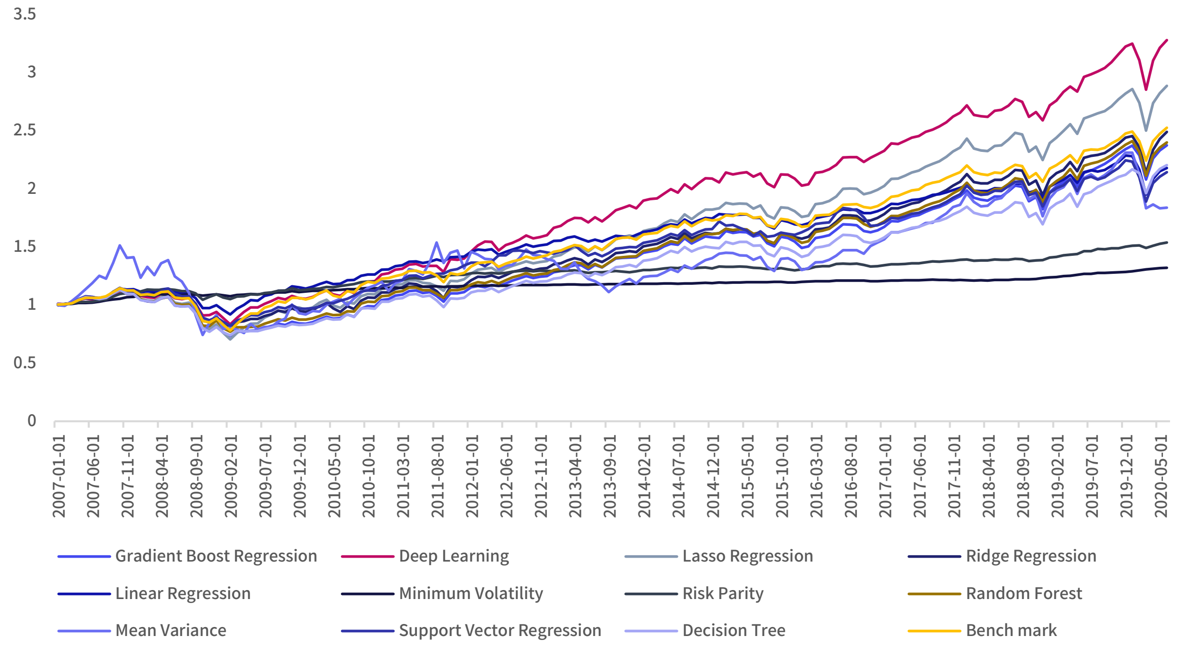
[Table 2] Application Data Set

Category	Ticker	Name	Description
Panel A : 7 Asset Class			
Equity	WIL5TMK	Wilshire 5000 Total Market	Equity
Equity	MSEMKF$	MSCI Emerging Markets USD	Emerging Country Equity
Bond	SPBDUSL	S&P US Treasury 20+ years	Treasury 20+Y
Bond	SPBDU10	S&P US Treasury 7-10 years	Treasury 7-10Y
Bond	SPBDUS3	S&P US Treasury 20+ years	Treasury 1-3Y
Commodity	GSCITOT	S&P GS Commodity Index	Commodity
Gold	GSGCTOT	S&P GS Commodity Index - Gold	Gold
Panel B : 11 Macro & Valuation Class			
Macro	BAAFF	Moody's Seasoned Baa Corporate Bond Minus Federal Funds Rate	Credit Spread
Macro	PCETRIM12M159SFRBDAL	Trimmed Mean PCE Inflation Rate	Inflation
Macro	GDPC1	Real Gross Domestic Product	GDP
Macro	WTISPLC	Spot Crude Oil Price : West Texas Intermediate(WTI)	WTI
Macro	M1	M1 Money Stock	M1
Macro	M2	M2 Money Stock	M2
Macro	T10Y2Y	10-Year Treasury Constant Maturity - 2-Year Treasury Constant Maturity	Duration Spread
Macro	UNRATE	Unemployment Rate	Unemployment
Macro	REAL_VOL	1-Year Rolling Volatility of S&P Index	Market Volatility
Macro	DFF	Effective Federal Funds Rate	Risk-free Rate
Valuation	S&PDIV	Dividend Yield of S&P Index	Valuation
Valuation	S&PPE	P/E of S&P Index	Valuation

Source : QRAFT Technologies, DataStream, Compustat, FRED

[Table 2] shows assets and macroeconomic data used for example analysis. There are four types of asset class data: stocks, bonds, commodities, and gold. In addition, 9 macroeconomics data and 2 valuation data are used. The in-sample period for the analysis is from January 1970 to December 2006, and the out-of-sample period is from January 2007 to June 2020. Using the parameters of each model estimated in the in-sample period, the weight of individual assets is calculated in the out-of-sample and the portfolio is constructed using predicted weights. The recent 10% of the in-sample period is used as validation data for parameter optimization. The label used for training used an objective function to maximize the Sortino Ratio for the next 12 months, and the target of portfolio volatility was set to be in between 6% and 17% per year. From the investor's point of view, upside volatility provides profit, while the downside the loss. It may be argued that the downside risk possesses true investment risk, therefore, Sortino Ratio may be closer to what investors would like to consider in the investment.

[Figure 13] Cumulative Return of Model Portfolios



Source : QRAFT Technologies, DataStream, Compustat, FRED

Machine learning models are well below the performance of the benchmark 60/40 portfolio, except the Lasso and DL models are outperforming. When comparing Lasso and DL, Lasso shows a non-consistent performance compared to BM, and the deep learning model shows steady excess return compared to BM.

[Table 3] Metrics of Model Portfolios

This table shows the performance of each portfolios over the sample period. The values in the table below are annual values, and in the case of Risk Free, 3-Year Treasury Bond were used. And, Bench-mark was set to 45% of U.S. Stocks, 15% of Emerging Market Stocks, and 40% of 10-Year Treasury bonds.

	Mean	Vol	Sharpe	Sortino	MDD
Panel A: Traditional Approach					
Bench-mark(60/40)	0.0647	0.0945	0.6846	0.9097	0.3173
Mean-Variance	0.0542	0.1862	0.2910	0.4253	0.5083
Minimum-Volatility	0.0120	0.0119	1.0141	2.8849	0.0169
Risk Parity	0.0238	0.0306	0.7774	1.1702	0.0794
Panel B: Machine Learning Approach					
Support Vector Regression	0.0523	0.0927	0.5637	0.7302	0.2824
Decision Tree	0.0554	0.1037	0.5344	0.7258	0.3466
Linear Regression	0.0519	0.0732	0.7086	0.9017	0.1941
Gradient Boosting Regression	0.0609	0.1026	0.5933	0.7841	0.3546
Random Forest	0.0612	0.0982	0.6234	0.8147	0.3205
Ridge	0.0644	0.1024	0.6284	0.8588	0.2922
Lasso	0.0761	0.1092	0.6971	0.9009	0.3743
QRAFT Deep Learning	0.0840	0.0938	0.8960	1.2679	0.2506

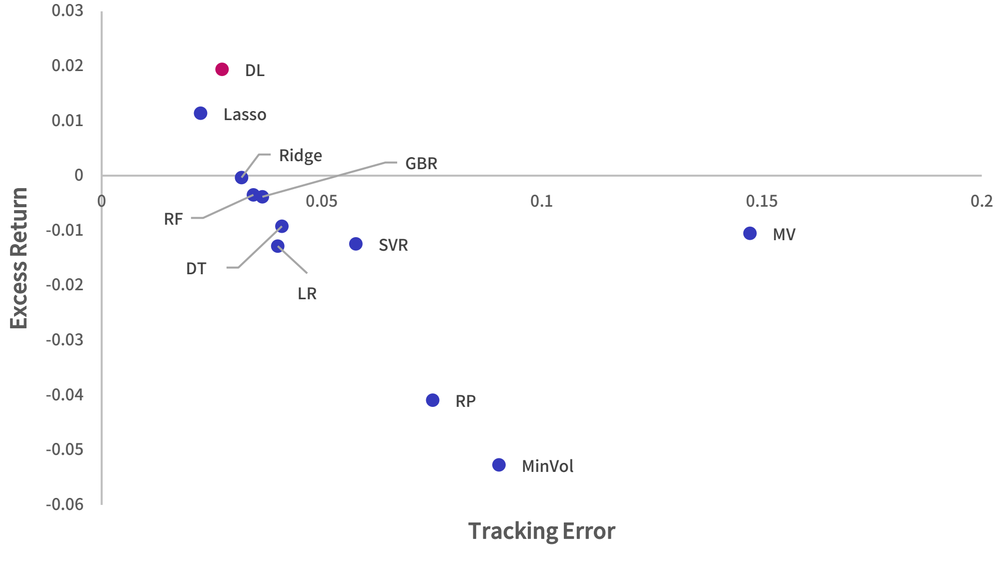
Source : QRAFT Technologies, DataStream, Compustat, FRED

[Table 3] shows the OOS back test results. Panel A is the result of traditional RP, MV and Min-Vol as well as benchmark portfolio. Panel B shows various ML models to the same learning environment to confirm the advantages of deep learning model.

First, all of Panel A's traditional approaches show results with average returns below the benchmark. On the other hand, Min-Vol and RP showed better indicators of risk-adjusted return while MV underperforms compared to BM. However, outperformance of Min-Vol and RP is due to low mean return and volatility. Therefore, it is difficult to argue that both methods have advantages over the BM when accounted for their return.

Looking at Panel B, other methods except the Lasso and DL methods have poor returns and risk-adjusted performance compared to the BM. Lasso shows the average annual return of 7.61%, higher 6.47% average return of BM. However, when reviewing the risk-adjusted indicators, Lasso proves lower to Benchmark in both the Sharpe and Sortino ratio, so it is difficult to argue that the model has outperformed out of sample. On the other hand, our DL has an annual average return of 8.40%, far above the benchmark mean return. Since Sharpe Ratio and Sortino Ratio are the highest among other ML models, it can be said to be superior to other ML methods.

[Figure 14] Excess Return & Tracking Error of Model Portfolios



Source : QRAFT Technologies, DataStream, Compustat, FRED

[Figure 14] is a graph showing excess returns and tracking errors of different models. Even if there are tracking errors with benchmark, it is relatively low and only the DL and Lasso shows excess returns. The rest of the traditional allocation and ML-applied portfolios do not show excess returns.

[Table 4] Information Ratio of Model Portfolios

	Mean return	Excess return	Tracking Error	Information Ratio
Panel A: Traditional Approach				
Bench-mark(60/40)	0.0647	-	-	-
Mean-Variance	0.0542	-0.0105	0.1473	-0.0712
Minimum-Volatility	0.0120	-0.0527	0.0903	-0.5831
Risk Parity	0.0238	-0.0409	0.0752	-0.5440
Panel B: Machine Learning Approach				
Support Vector Regression	0.0523	-0.0124	0.0578	-0.2150
Decision Tree	0.0554	-0.0092	0.0410	-0.2256
Linear Regression	0.0519	-0.0128	0.0400	-0.3198
Gradient Boosting Regression	0.0609	-0.0038	0.0365	-0.1047
Random Forest	0.0612	-0.0035	0.0345	-0.1004
Ridge	0.0644	-0.0003	0.0318	-0.0095
Lasso	0.0761	0.0114	0.0225	0.5089
QRAFT Deep Learning	0.0840	0.0194	0.0274	0.7079

Source : QRAFT Technologies, DataStream, Compustat, FRED

[Table 4] shows the information ratio[8] of each portfolio. The value obtained by dividing the excess return by the tracking error is called the information ratio and is a measure of how the portfolio performs better than the benchmark consistently. The information ratios of all methods except DL and Lasso produces negative values. On the other hand, the information ratio of DL is 0.7079, which is larger than that of Lasso (0.5089). This shows that even if each strategy is evaluated by IR, the DL method has advantages.

[Table 5 : Significance Test for Model Portfolio's Excess Returns]

The table shows the excess return versus the benchmark for each portfolio over the out of sample period. The values in the table below are monthly basis, and Newey and West (1987) t-statistics with a maximum lag of 12 are used.

	Coefficient	Std. Error	t-value	p-value
Panel A: Traditional Approach				
Mean-Variance	-0.0047	0.0018	-2.6382	0.0083
Minimum-Volatility	-0.0042	0.0039	-1.0832	0.2787
Risk Parity	-0.0056	0.0022	-2.5631	0.0104
Panel B: Machine Learning Approach				
Support Vector Regression	-0.0014	0.0015	-0.9030	0.3665
Decision Tree	-0.0009	0.0015	-0.5909	0.5546
Linear Regression	-0.0016	0.0010	-1.6773	0.0935
Gradient Boosting Regression	-0.0003	0.0014	-0.2312	0.8172
Random Forest	-0.0002	0.0013	-0.1782	0.8585
Ridge	-0.0003	0.0011	-0.2774	0.7815
Lasso	0.0015	0.0006	2.5590	0.0105
QRAFT Deep Learning	0.0021	0.0009	2.2936	0.0218

Source : QRAFT Technologies, DataStream, Compustat, FRED

[8] Bacon, 2008, Practical Portfolio Performance Measurement & Attribution, Wiley

It is also necessary to examine whether the magnitude of the excess return generated by each model is statistically significant. We conduct Newey and West (1987) T-test to review the significance of excess returns. Looking at Panel A of [Table 5], in the traditional method, all methods show negative excess returns, especially in the case of MV and RP, statistically significant negative value.

Panel B shows that the excess returns from models except Lasso and DL are not statistically significant. Even if the portfolio returns temporarily deviate from the benchmark, the significance cannot be statistically verified. On the other hand, DL model portfolio exhibits a statistically significant positive value, and its value is larger than that of Lasso. The results are consistent with the performance trends shown in [Table 4] and proves the advantage of DL portfolio.

If the back-test period are longer, there is a possibility that the investment return measured for the whole period may be distorted by some specific return period. Therefore, it is important to look at the distribution of the returns with given investment horizon. In this paper, we examine the probability that each portfolio's rolling return exceeds the benchmark.

[Table 6] Rolling return Win Ratio of Model Portfolios

	3-month	6-month	12-month	24-month
Panel A: Traditional Approach				
Mean-Variance	0.2875	0.2420	0.2119	0.1379
Minimum-Volatility	0.5438	0.6178	0.5695	0.5931
Risk Parity	0.3000	0.2484	0.2119	0.1586
Panel B: Machine Learning Approach				
Support Vector Regression	0.4500	0.4522	0.4503	0.3931
Decision Tree	0.4813	0.4650	0.5166	0.5517
Linear Regression	0.3250	0.2803	0.2583	0.2552
Gradient Boosting Regression	0.5375	0.5414	0.6093	0.6000
Random Forest	0.5313	0.5860	0.5960	0.6483
Ridge	0.5625	0.5478	0.6556	0.7172
Lasso	0.6688	0.7134	0.7881	0.8207
QRAFT Deep Learning	0.6063	0.7325	0.8344	0.8897

Source : QRAFT Technologies, DataStream, Compustat, FRED

[Table 6] shows the calculated win ratio by calculating the portfolio's rolling return against the benchmark. For DL model, the winning rate of 24 months rolling return is 88.97%, and 60.63% for 3-month rolling return. It shows that as the investment horizon expands DL model is more likely to outperform the BM. Lasso model, on the other hand, while the win rate of the 3-month rolling return is 66.88%, which is higher than that of the DL model, it is less likely to maintain its return as investment period becomes longer. Therefore, we can confirm the superior returns provided by the DL model since the increase in chance for outperformance for longer periods are significantly larger than other investment methods.

Conclusion

Various investor and different markets have actively utilized several asset allocation strategies. Rather than blindly pursuing high returns, it's important to consider the risks involved in your portfolios. In recent years, strategies like mean-variance and risk parity have widely been adapted, away from the traditional asset allocation strategies like the simple 60/40 portfolio. These strategies suggest that past time series is indicative of the future, which shows a clear limitation as an investment strategy when considering market dynamics. In order to address these issues and effectively allocate assets, we propose utilizing a deep learning technique.

This paper presented a simple model structure and the justification of asset allocation strategy using the currently available deep learning models. For empirical analysis, the results were compared to the benchmark traditional asset allocation strategy, 60/40 Portfolio. In addition, the advantages of the deep learning model were confirmed by comparing the performance with other relatively well-known machine learning models. When examined using various measures, the performance of our deep learning Asset Allocation Model was the most dominant.

However, despite this achievement, it is difficult to argue that other machine learning methodologies are not as effective. One of the reasons is that the model performance of out of sample data in both the machine learning and deep learning depend largely on how the in-sample data and parameters are set. In order to consider this point, the rolling-window method that continuously re-learns over time and reuses the results of the re-trained model should be examined. While the above process is omitted in this paper, the asset allocation model that we commercially use constantly updates the model and incorporate more recent data and structures. This leads to achieving a more realistic and better performance of the model by continuously merging better methods and data. Future research will cover the topic with various analyses.

Establishing and implementing investment strategies using financial data can be difficult. But we try to solve these difficulties by utilizing a wide range of effective AI technologies. To be precise, the use of deep learning in asset allocation is not entirely a new domain, but just an extension of traditional investment methods. Traditional asset allocation strategies assume and depend on the continuity of the statistical stability while the All-Weather strategy differs in incorporating different market environments. They all fall under the same basket with a deep learning model for utilizing historical financial data. In this sense, deep learning is only a presentation of a new direction and not the whole new plane.

Disclaimer / Notice

* This Paper is not intended as an offer or solicitation for the purchase or sale of any financial instrument such as fund. Also, The views contained herein are not intended as a recommendation of particular securities, financial instrument or strategies to particular clients.
* The information used herein has been obtained from sources believed to be reliable, but neither QRAFT Technologies nor its affiliates warrant its completeness or accuracy. The recipient of this report must make its own independent decision regarding any security or financial instrument mentioned herein. Therefore, this paper can not be used as evidence of legal liability for investment results under any circumstances.
* This Paper and all of the information contained in it, including without limitation all text, data, graphs, charts is the property of QRAFT Technologies. The information may not be modified, reverse-engineered, reproduced in whole or in part without prior written permission from QRAFT Technologies.

QRAFT Technologies aims to maximize efficiency in investment by minimizing inefficient costs in traditional asset management by utilizing AI technology from lowering the cost of finding alpha to lowering execution costs.