

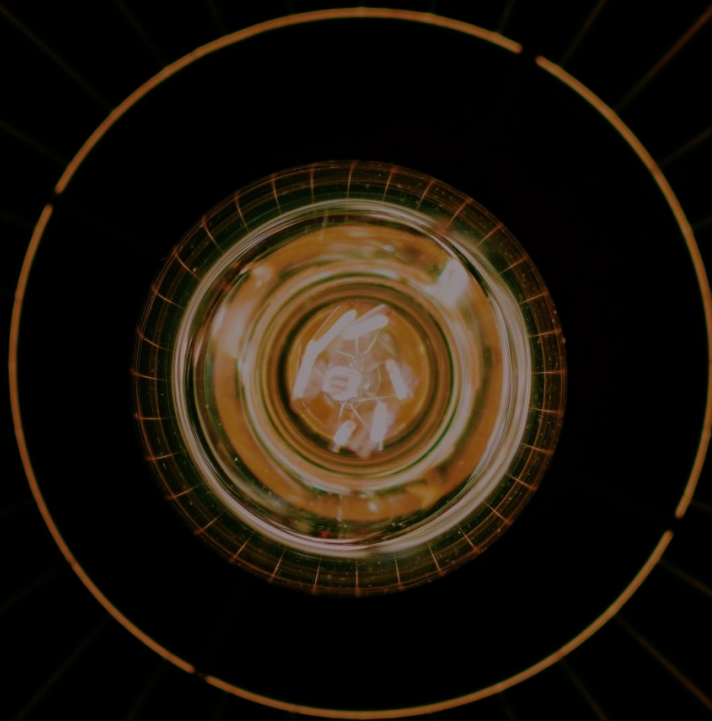
**QRAFT AI Quant Series**

# **QRAFT Deep Learning Asset Allocation**

---

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

**August  
2020**



# Contents

요약	3
제안 배경	4
QRAFT Deep Learning 모델 구조	11
실증 분석	16
결론	21

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

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

# 요약

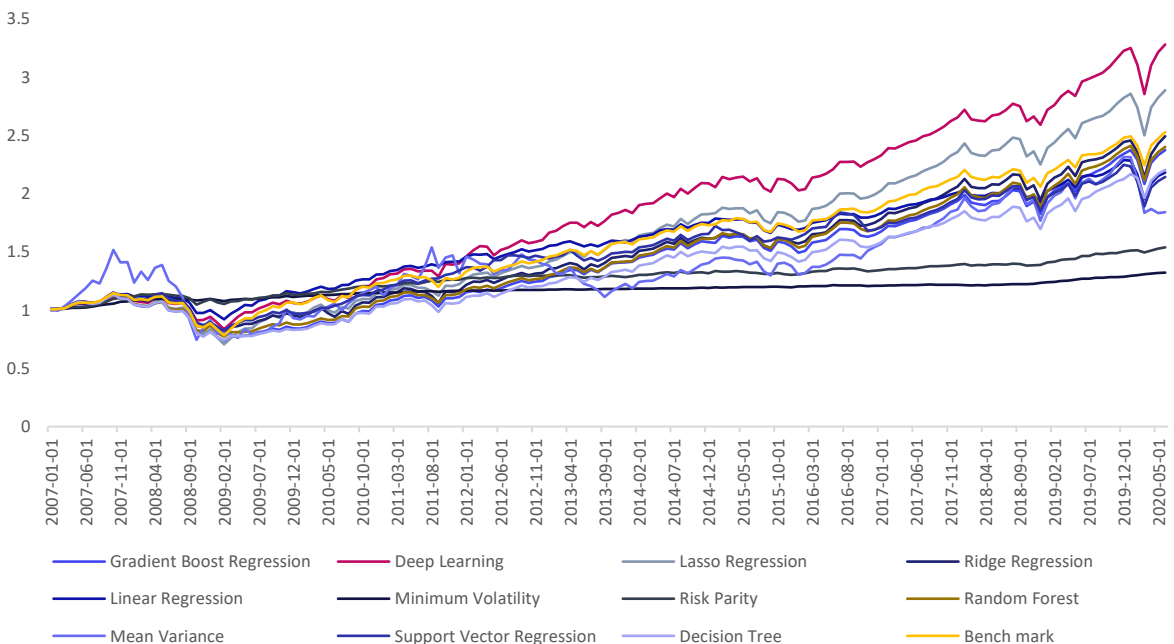
자산배분은 투자가 가능한 자산 군들을 적절히 배분하여 포트폴리오를 구성하는 것을 의미한다. 자산배분을 통해 상대적으로 높은 위험 군의 자산인 주식부터 낮은 위험 군의 자산인 채권 및 현금까지 골고루 포트폴리오에 담아 포트폴리오 전체의 위험을 크게 줄일 수 있다. 꾸준한 투자 수익을 얻기 위해서는 수익률을 높이는 것도 중요하지만 위험을 줄이는 것 또한 매우 중요한데, 당사의 딥러닝을 활용한 자산배분은 기존 잘 알려진 벤치마크와 동일한 위험 대비 더 높은 수익을 이끌어 내는 자산배분 모델을 제시하는데 그 의의가 있다.

전통적인 자산배분에 의한 투자는 실제 많은 자산운용사에서 펀드 형태로 구성되어 활용되고 있고, 개인들 또한 어렵지 않게 포트폴리오를 구성해 활용 중에 있다. 하지만, 이러한 전통적인 포트폴리오 구성 방법은 예측 모델을 의미하지 않고, 금융 시장은 역사적 분포의 연속성(Continuity)이 있다고 보기가 어렵다. 또한, 포트폴리오 성과를 위한 Momentum 전략, Long-Term Reversal 전략과 같은 전술적 자산배분을 시행한다고 하더라도, 이러한 전략들은 최적화되어 있다고 보기가 어렵다. 당사는 이러한 문제를 해소하고, 더 나은 전략의 성과를 위해 인간이 일일이 테스트를 진행하는 오류를 범하는 것이 아닌 인공지능 기술을 활용한다. 당사의 기술을 통해 이러한 퀀트 전략들은 최적화 단계에 누구보다 빠르게 도달할 수 있으며, 효율적 배분과 동적인 시장상황에 유연한 대처가 가능한 알고리즘을 통해 리스크 대비 더욱 높은 수익을 이끌어 낼 수 있다.

간단한 실증 분석을 통해 딥러닝을 통한 자산배분의 성과를 점검해본 결과, 분석의 Base Model인 전통적 자산배분 전략인 60/40 포트폴리오와 Risk-Parity 전략 대비 더욱 높은 성과를 보였다. 또한, 딥러닝과 큰 흐름을 함께하는 머신 러닝 방법론들과 비교해본 결과도 딥러닝을 통한 자산배분이 가장 좋은 성과를 보임을 확인하였다.

금융 데이터들을 활용하여 투자 전략을 수립하고 집행하는데 있어 많은 어려움에 직면할 수 있다. 당사는 이를 인공지능을 통해 그 과정을 효율적으로 해소해 나갈 뿐 아니라, 더 좋은 성과를 도출해내고자 한다. 전통적 자산배분 전략, Risk Parity 전략, All Weather 전략 그리고 Deep Learning을 활용한 전략 모두 과거 금융 데이터를 활용한다는 점이라는 큰 골자 안에서는 Deep Learning과 함께한다고 볼 수 있다. Deep Learning을 활용한 자산배분 전략은 기술의 발전에 따라 그 프로세스의 간소화 및 성과 개선의 여지가 뚜렷하다는 점에서, 시장의 변화를 이끌어 낼 것으로 기대하고 있다.

## Portfolio Performance



자료: QRAFT Technologies, Compustat, DataStream, FRED

## 제안 배경

자산배분은 성공적인 투자 전략의 요소 중 하나이다. 투자가 가능한 자산들은 최근 더욱 다양하다. 자산배분은 이러한 투자가 가능한 자산군들을 적절히 배분하여 포트폴리오를 구성하는 것을 의미한다. 자산배분을 통해 상대적으로 높은 위험 군의 자산인 주식부터 낮은 위험 군의 자산인 채권 및 현금까지 골고루 포트폴리오에 담아 포트폴리오 전체의 위험을 크게 줄일 수 있다. 꾸준한 투자 수익을 얻기 위해서는 수익률을 높이는 것도 중요하지만 위험을 줄이는 것 또한 매우 중요하다. 당사의 자산배분의 궁극적인 목표는 기존 잘 알려진 벤치마크와 동일한 위험 대비 더 좋은 수익을 이끌어 내는 자산배분 모델을 제시하는데 그 의의가 있다. 따라서, 자산배분을 통해 위험을 줄이는 것에도 중점을 둘 필요가 있다. 단순히 수익 관점으로 접근하게 되면 특정 자산군에 너무 많은 비중이 할당되어 적절한 위험 분산이 어렵다.

### 1. Traditional Asset Allocation

전통적 자산배분 전략의 방법으로 60/40 포트폴리오가 존재한다. 이는 주식 60%와 채권 40%로 이루어진 포트폴리오를 구성하는 것으로, 단순 하나의 자산만 보유하는 것 보다 위험 측면에서 더욱 강건한 것으로 잘 알려져 있다. 특히 위 방법의 경우 투자자가 누릴 수 있는 훌륭한 분산의 효과에 비해 구성법이 매우 간결하기 때문에, 개인 투자자들에게도 쉽게 권할 수 있는 투자방법이다.

Markowitz(1952)<sup>1</sup>는 포트폴리오 구성 시, 개별 자산의 Risk-Return뿐만 아니라 개별 자산과 전체 포트폴리오 간의 리스크-리턴 관계를 고려해야 한다는 주장을 하며 현대 포트폴리오 이론을 개척하였다. 주어진 자산군의 과거 수익률과 공분산행렬을 통해 특정 리스크 내에서 구성할 수 있는 최대 기대수익률 비중을 구성할 수 있으며(Efficient Frontier), 아래 첫번째 식과 같이 한계 리스크와 수익률의 증분 값을 이용해 위험 대비 수익률이 최대치가 되는 Mean-Variance 포트폴리오 비중을 구할 수 있다. 또한, 주어진 자산군들의 분산과 공분산을 이용해 아래 두번째 식과 같이 전체 포트폴리오의 분산을 최소로 만드는 비중을 구함으로써, Minimum-Volatility 포트폴리오를 구성할 수 있다.

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

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

$\mu$  : 기대수익률

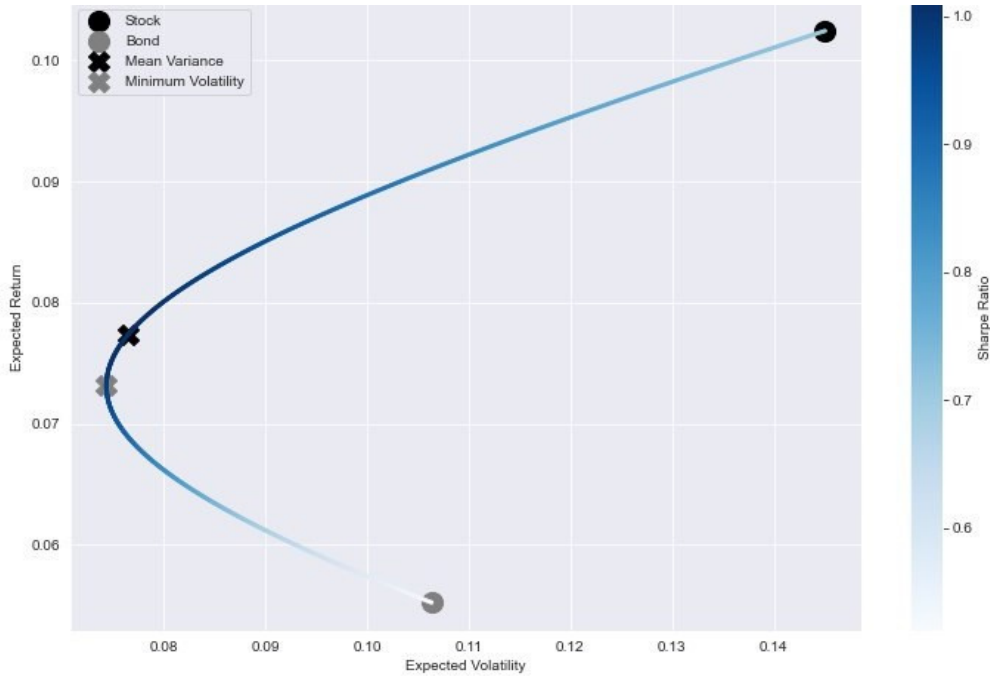
$w$  : 개별자산 비중

$\Sigma$  : 자산들의 공분산

$r_f$  : 무위험 수익률

<sup>1</sup> Harry Markowitz, 1952, Portfolio Selection, The Journal of Finance 7-1, 77-91

[Figure 1] Risk & Returns Relationship



자료: QRAFT Technologies, Compustat

포트폴리오의 위험의 측면에서 접근한 자산배분 전략으로 리스크 패리티(Risk Parity) 전략이 있다. 리스크 패리티 전략은 포트폴리오 구성 시 편입되는 자산군들의 위험을 동일한 수준으로 맞춰주는 전략이다. 편입된 자산군들의 위험 수준을 동일하게 맞추므로써 전체 운용 성과에 미치는 영향이 자산군 별로 거의 동일 해진다. 주어진 포트폴리오에서 포트폴리오의 리스크는 개별 자산의 위험 기여도와 개별 자산의 비중의 곱의 합으로 표현될 수 있다.

$$\text{개별자산의 위험 기여도: } \sigma_i(w) = w_i \times \frac{\partial \sigma}{\partial w_i} = w_i \times \frac{(\Sigma w)_i}{\sqrt{w' \sigma w}}$$

$$\text{개별 자산의 한계 위험 기여도: } \frac{\partial \sigma}{\partial w_i} = \frac{\sigma_{i,P}}{\sigma} = \rho_{i,P} \times \sigma_i$$

$$\text{포트폴리오 총위험: } \sigma_p = \sum_{i=1}^N w_i \times \frac{\partial \sigma}{\partial w_i}$$

$w$ : 개별자산 비중

$\sigma$ : 포트폴리오 수익률의 표준편차

$\rho$ : 상관계수

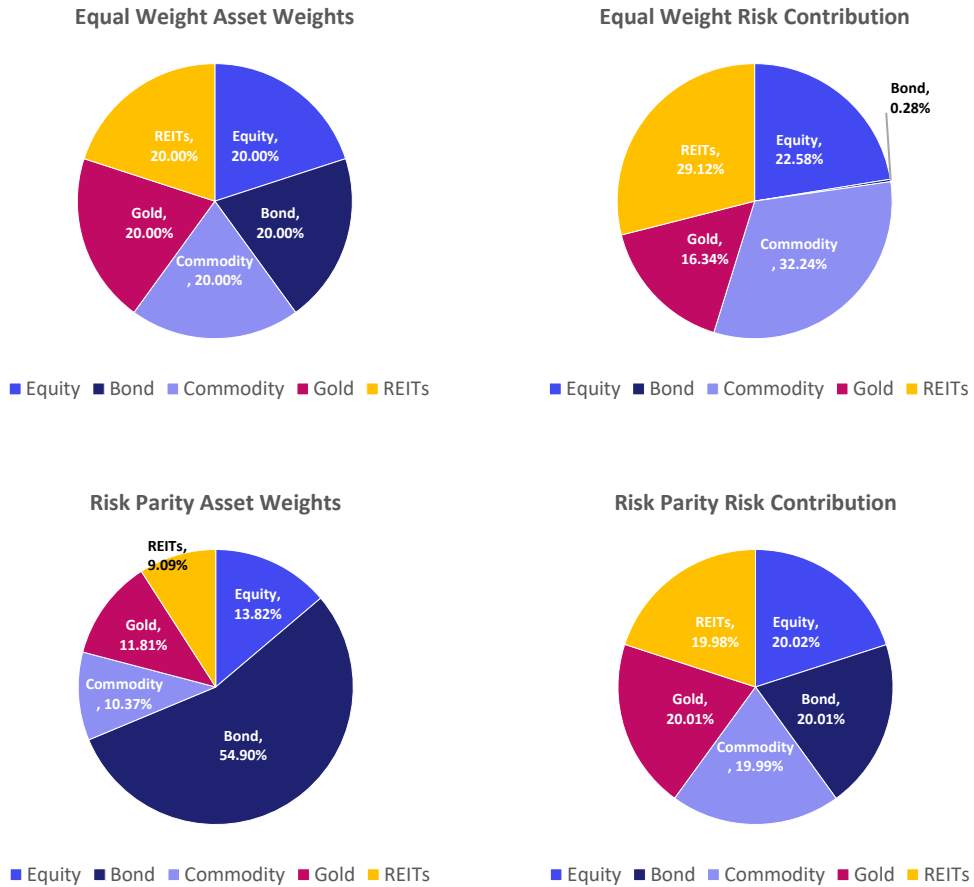
$i$ : 개별 자산

$P$ : 포트폴리오

리스크 패리티 전략은 포트폴리오 구성을 위해 개별 자산이 포트폴리오 전체에 기여하는 위험이 같도록 하는 비중 벡터를 찾는다. 아래 식의 목적함수를 도출하고, 최적화를 통해 리스크 패리티를 만족하는 자산의 비중을 도출한다.

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

[Figure 2] 리스크 관점에서의 포트폴리오 노출



자료: QRAFT Technologies, DataStream

위 [Figure 2]는 총 5개의 자산을 동일 가중과 Risk Parity로 배분한 결과를 나타낸다. 포트폴리오는 주식, 채권, 원자재, 골드, 리츠로 이루어져 있다. 먼저 동일 비중의 경우 포트폴리오 내에서 주식의 비중은 20%이지만 포트폴리오 위험에 기여하는 비중은 22.58%로 더욱 높아지며, 원자재의 경우 32.24%를 확인할 수 있다. 반면 Risk Parity의 포트폴리오 위험 기여도를 보면 전체 5개 자산군이 동일하게 나타나는 것을 관찰할 수 있다. 즉, 동일 가중으로 배분한 포트폴리오의 위험은 그 위험 수준이 채권과 같은 안전자산이 아닌 대부분 위험 자산에 의해 결정되어 버려 적절한 위험 배분으로 보기가 어렵다. Risk Parity 전략은 모든 자산군들이 포트폴리오에 기여하는 위험을 같도록 만들어 줌으로써, 포트폴리오의 하방 위험을 크게 줄일 수 있다.

기존 포트폴리오의 비판을 통해 등장한 All Weather Portfolio 또한 리스크 패리티 전략 중 하나이다. The All Weather Strategy(BridgeWater, 2009)<sup>2</sup>는 대부분 기관들의 포트폴리오의 문제점은 분산화의 부족으로 인해 비효율적인 포트폴리오를 이끌어 내는데, 이러한 포트폴리오는 그들이 안고 있는 위험 대비 너무 적은 수익률만을 안겨준다고 주장한다. 즉, 대부분의 투자자들은 자산군 클래스의 분산화에 대한 이점은 충분히 인지하고 있으나, 리스크가 아닌 자본의 배분에 더욱 초점을 둔다는 것이다. 하지만 역사적 변동성을 기반으로 포트폴리오의 위험을 동일하게 하는 기존 리스크 패리티 전략은 상관관계 및 변동성에 바탕을 두고 있어 경제 상황에 따라 매우 민감하게 반응할 수밖에 없다. 이러한 이유로는 자산군 간의 상관 관계는 일정하지 않으며, 리스크 또한 예측하기가 매우 어렵기 때문이다. 이 점을 타파하고자 All Weather 전략은 각기 다른 경제 환경에서의 자산군들의 구조적 관계를 고려하였고 예상치 못한 경제 상황의 변화로 인한 영향을 최소화한다.

<sup>2</sup> BridgeWater, 2009, The All Weather Strategy

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

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

자료: QRAFT Technologies

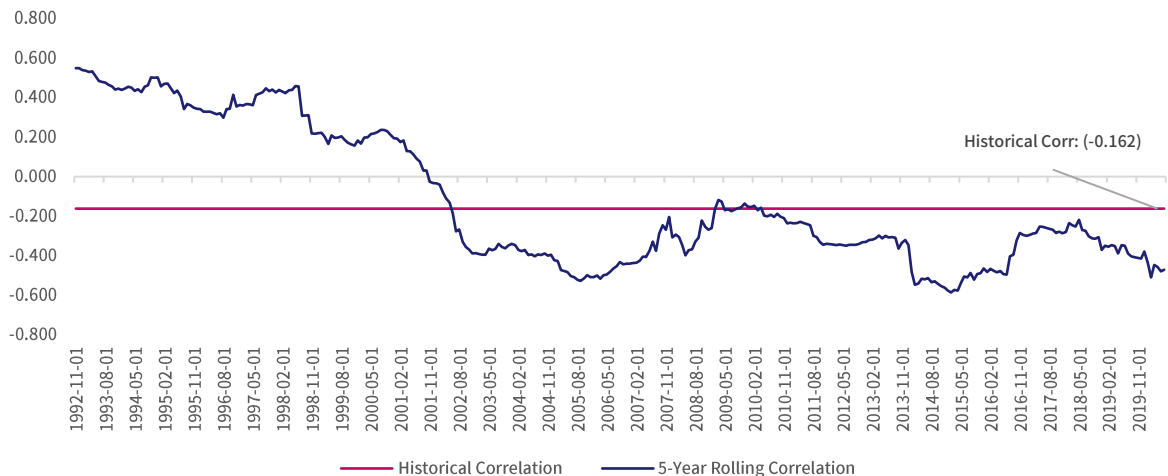
위 [Table1]은 All Weather 전략에서 어떻게 각기 다른 거시경제 환경이 각 자산군들에게 영향을 미치는지에 대해 보여준다. 4개의 시장 환경에서 리스크 균형을 맞추므로써, 어느 하나의 환경에 대한 포트폴리오의 민감도를 최소화하여 일관적으로 자산군의 리스크 프리미엄을 이끌어낼 수 있고, 기존 전통적 자산배분 및 리스크 패리티 전략 대비 더욱 강건하게 포트폴리오를 구성할 수 있다.

## 2. Why Deep Learning?

전통적인 자산배분에 의한 투자는 실제 많은 자산운용사에서 펀드 형태로 구성되어 활용되고 있고, 개인들 또한 어렵지 않게 포트폴리오를 구성해 활용 중에 있다. 하지만, 이러한 전통적인 포트폴리오 구성 방법은 예측 모델을 의미하지 않는다. 전통적인 자산배분법은 과거의 데이터 분포가 미래에도 그럴 것이라는 역사적 분포의 연속성(Continuity)을 가정으로 갖고 있기 때문에, 우리가 알고 있는 금융시장과 비교해 볼 때 현실적으로 보기가 어렵다.

예를 들어, 일반적으로 주식과 채권 사이의 상관관계는 음(-)의 상관관계를 지닌다고 생각하지만, 두 자산 간의 관계는 아래 [Figure 3]과 같이 유동적인 변화를 거쳐 왔다. 이처럼 시간의 변화에 따라 달라지는 관계에 적절히 대응하여 포트폴리오를 구성하여야만이 자산배분의 효과를 극대화할 수 있다.

[Figure 3] S&P500과 US Treasury Note의 5년 Rolling Correlation

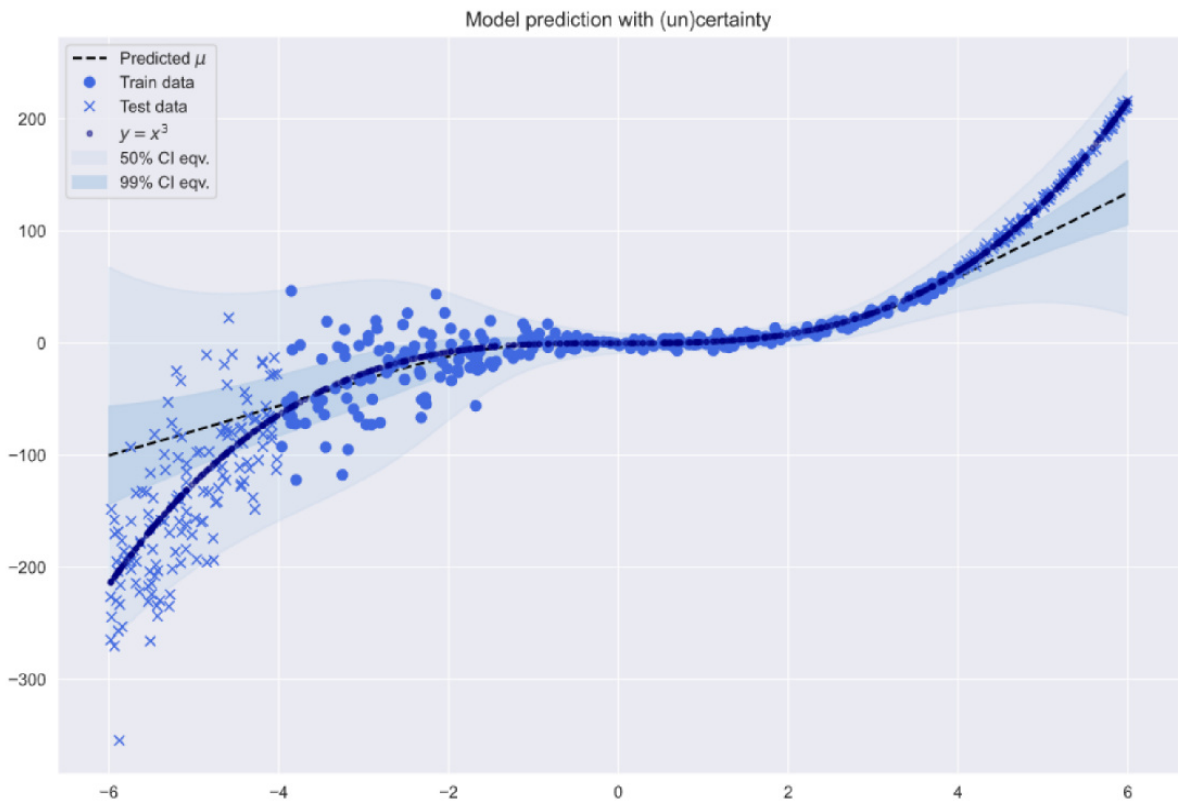


자료: QRAFT Technologies, Compustat

포트폴리오 성과를 위한 Momentum 전략, Long-Term Reversal 전략들을 활용하는 것을 전술적 자산배분(Tactical Asset Allocation) 이라고 한다. 하지만, 이때도 Momentum값을 측정하기 위한 최적의 기간(Optimal Period)은 무엇이며, Long-Term Reversal을 판단하는 기준은 무엇인가에 대한 난관에 봉착한다. 이러한 기준은 선행 연구들에 의해 결정되는 경우가 많고, 일일이 테스트를 해보기에는 많은 리소스가 할당되어야만 하는 문제가 존재한다. 이러한 문제들을 해소하고, 더 나은 전략의 성과를 위해 당사는 인간이 일일이 테스트하는 오류를 범하는 것이 아닌 인공지능 기술을 활용한다. 잘 알려진 퀀트 전략들은 이러한 과정을 통해 최적화 단계에 누구보다 빠르게 도달할 수 있으며, 효율적 배분과 동적인 시장 상황에 유연한 대처가 가능한 알고리즘을 통해 리스크 대비 더욱 높은 수익을 이끌어 낼 수 있다.

Olalekan(2016)<sup>3</sup>에 따르면 딥러닝은 비선형적 관계를 포착할 수 있다. 시계열 모형 중 자기상관모형이나 이동 평균 모형과 같은 통계학적 모델링을 적용할 시, 그 Window 값을 얼마나 주어야 제대로 모델링이 가능한지를 시계열 모형의 차수에 반영한다. 모형의 차수 추정을 위해 여러 통계적 기법이 활용되어지나 대부분 선형 모형에 기반한 통계학적 시계열 모형들은 매우 복잡한 비선형적 관계를 지닌 데이터를 다루는데 한계가 존재한다.

[Figure 4] Model Prediction with (un)certainty



자료: QRAFT Technologies

위의 [Figure 4]는 당사의 딥러닝을 통해 비선형적 관계의 데이터 예측 모델의 예시이다. 대부분의 금융 데이터 간에는 비선형적 관계가 존재하고, 이를 선형으로 단순 추정할 시 선불리 그 관계를 오인할 수 있다. 따라서 금융 데이터를 활용 시, 비선형적 관계를 추론할 수 있는 딥러닝이 강점을 가진다.

<sup>3</sup> 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

Deep Learning의 기본적인 구조인 Neural Network(이하NN)는 유연한 구조 설계를 큰 장점으로 꼽을 수 있다. 과거 데이터 분석의 연속성의 한계 및 시장 상황에 대한 유연한 대처를 위해 적절히 사용될 수 있음을 의미한다. 또한 전통적인 머신 러닝 방법론들은 특정 Task에 맞춰진 알고리즘이 구현되는 데 반해 NN은 레이어의 구조와 활성화함수 등을 자유롭게 구성할 수 있고 아웃풋 레이어 구조에 따라 회귀, 분류, 차원 축소, 분포 추정, 데이터 생성(샘플링) 등 여러 목표를 수행할 수 있다. 이러한 유연성은 전략을 설계하는 모든 과정에서 적절히 활용되며, 전체적인 프로세스에 들어가는 리소스를 크게 줄인다. 학습 과정에서는, 최근 데이터와 함께 계속적인 학습을 이루어 내고 횡단면 매크로 데이터와 자산군들의 데이터 들로 추출된 특징들을 바탕으로 팩터 Formulation을 최적점에 이르도록 한다. 하지만, 이러한 Deep Learning을 활용하는데 있어 사전적으로 해소되어야 할 몇 가지 문제점이 존재한다. 크게 총 3가지로, 당사가 제안한 문제점은 충분한 표본 데이터의 부족, 데이터 노이즈, 오버 피팅 문제이다.

#### (1) Sample Data Deficit

현재 자산배분 모델은 레이블 데이터를 학습한 최적화된 포트폴리오를 도출하는 것을 그 목적으로 한다. 레이블 데이터를 학습하기 위해선 2가지 선제 조건이 필수 불가결한데, 그것은 ‘유의미’하고 ‘충분한’ 데이터가 존재해야만 한다는 것이다. ‘유의미’하다는 것의 의미는 자산군 레벨에서의 베타, 모멘텀, 평균 회귀와 같은 팩터의 계산은 어떻게 이루어져야만 하나라는 것이다. 당사는 이러한 유의미함을 이끌어 내기 위해 딥러닝 모델이 모멘텀 값을 계산하도록 학습시키는 방식을 채택한다. 다음으로 얼마나 ‘충분하냐’의 문제에 직면한다. 모델의 성능을 이끌어 내기 위해서는 충분한 표본 데이터가 요구되는데, 금융 데이터들은 활용할 수 있는 Feature 대비 너무 짧은 시계열만이 존재한다. 가령 40년 데이터의 경우 월별로 480개의 아주 작은 Sequence만이 활용 가능하다. 이럴 경우 차원의 저주(The Curse of Dimensionality)와 같은 문제점에 직면할 수 있다. 특히나 자산배분은 장기적인 시계열 데이터가 필요하다. 대략적으로 최소 1980년도부터의 데이터가 Training Set에서 사용되어야 하며, 장기적인 경제 사이클, 단기적인 경제사이클, 단기적인 팩터 현상 등을 동시에 고려할 수 있어야만 한다. 존재하지 않는 데이터는 MICE Imputation을 활용하거나, NaN Embedded Layer<sup>4</sup>를 활용한다. 본 페이지에서는 MICE Imputation을 활용한 결과를 제시한다.

#### (2) Data Noise

$$AR(1): P_t = \alpha P_{t-1} + \varepsilon_t$$

$$P_t = P_{t-1} + I_{t-1} + \varepsilon_{t-1}$$

$$P_t - P_{t-1} = I_{t-1} + \varepsilon_{t-1}$$

$$P_t : \text{Asset Price}_t$$

$$I_t : \text{Information}_t$$

$$\varepsilon_t : \text{Noise}_t$$

두번째로, 데이터의 노이즈 문제이다. 시계열 Feature 자체에는 노이즈가 존재한다. 위, AR(1) 모형에서 다음 주가는 현재주가+정보+노이즈에 의해 결정된다고 할 때, 여기서 딥러닝 모델이 포착해야 하는 건 정보이다. 하지만, 실제로 노이즈가 차지하는 요소는 정보량이 차지하는 요소보다 크다. 노이즈로 인해 상대적으로 정보를 포착해낼 수 없으며, 결과적으로 다음 값에 대한 최선의 예측 값은 현재 값이 되어버린다. 따라서 노이즈를 효과적으로 제거하여 정보를 얻는 것이 중요하다. 노이즈를 제거하기 위한 방법으로 전통적인 노이즈 제거 방법인 Moving Average(MA, EMA 등)과 Bilateral Filter를 활용한다(Sylvain Paris, 2009<sup>5</sup>).

<sup>4</sup> QRAFT Technologies에서 독자적으로 개발한 Neural Network로 금융데이터에 만연한 Not Available Value들을 효과적으로 마스킹하고, 필요한 정보만을 Auto-Scale하여 Embedded Feature로 Mapping한다. 본 방식을 통해 Forward-Fill이나 Masking을 하지 않고도 누락된 데이터를 쉽고 효과적으로 처리할 수 있다.

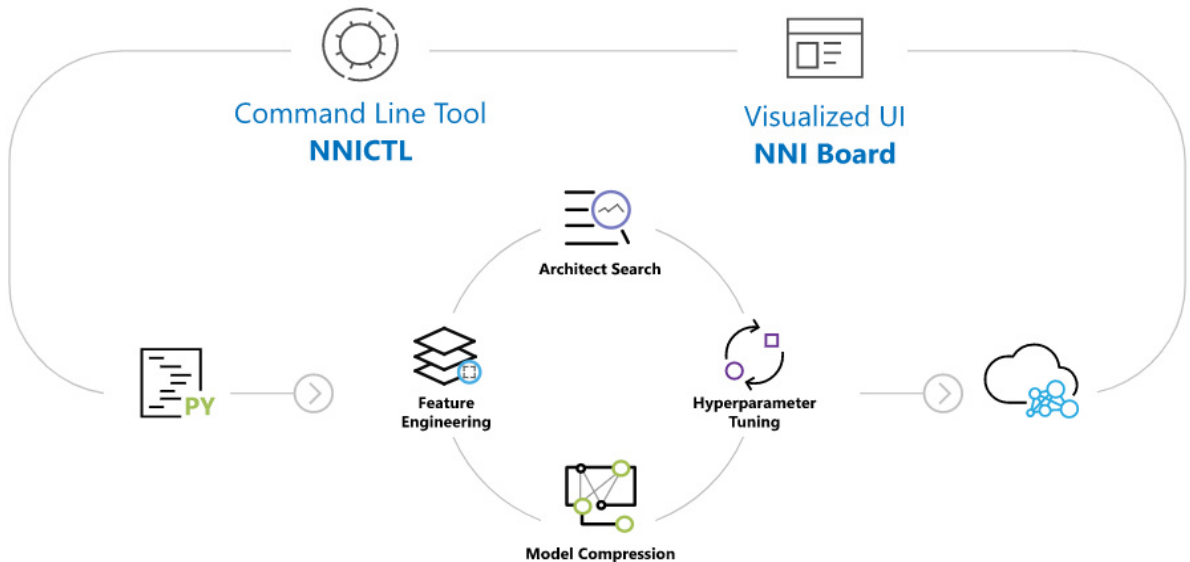
<sup>5</sup> Sylvain Paris, Pierre Kornprobst, Jack Tumblin and Fredo Durand, 2008, Bilateral Filtering: Theory and Application, Foundations and Trends in Computer Graphics and Vision 4-1, 1-73

그리고, 데이터의 정상성(Stationary) 검증을 위해 ADF-Test를 활용한다. 정상성이 없다면 시계열 학습 과정에서 오인된 상관관계를 도출할 가능성이 높아진다. 이러한 고전적인 시계열 노이즈 제거 방법은 제거하지 않은 데이터에 비해서 더 나은 학습 개선 효과를 일으킨다. 하지만, 이러한 고전적인 노이즈 제거 방법은 학습이 일어나기 이전 Input 데이터를 사전적으로 처리해준 후 학습을 진행해야만 한다. 따라서, 학습과정에서 자동적으로 노이즈를 제거하기 위해 CNN Stacked Auto Encoder<sup>6</sup> 기반의 Denoising Module 또한 활용한다.

### (3) Over-Fitting

마지막은 Over-Fitting의 가능성이다. 머신 러닝의 경우 하이퍼파라미터의 수가 매우 많이 존재하며, 하이퍼파라미터의 설정을 어떻게 하는가에 따라 모델의 in-sample 성능과 out-of-sample 성능의 차이가 현저하게 벌어질 수 있다. 따라서 네트워크의 하이퍼파라미터를 최소화하며, 네트워크 구조를 최적화를 하는 것이 Over-Fitting을 방지하는 데 도움이 된다. 이를 위해 Microsoft사의 NNI를 사용하여, 미리 정해진 하이퍼파라미터의 상태 공간을 탐색 및 최적의 구조를 자동화한다.

[Figure 5] NNI Flow



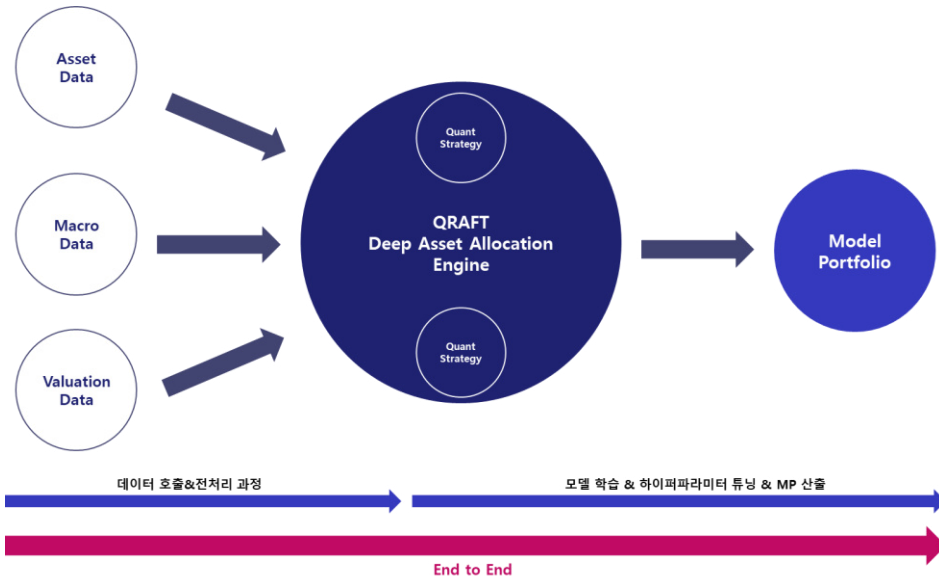
자료 : Microsoft

특히나 수익률 데이터는 데이터의 노이즈로 인해 모델이 잘못 학습하여 Over-Fitting 가능성이 매우 높다. 따라서 당사의 자산배분 모델은 기존의 주가 예측, 자산군 수익률 예측과 같은 직접적인 수익률 예측 모델과는 다른 방식을 적용한다. QRAFT Deep Learning Asset Allocation Model은 최적 포트폴리오를 따로 수치적으로 계산한 후에 이에 대한 레이블을 학습하고, 간접적으로 수익률을 예측하도록 유도한다. 이러한 학습 방법을 통해 Over-Fitting 가능성은 크게 줄어들고, 딥러닝 모델을 더욱 잘 활용할 수 있도록 한다.

<sup>6</sup> CNN(Convolutional Neural Network) 기반의 Auto Encoder 알고리즘으로, Stacked Auto Encoder는 여러 개의 히든 레이어를 가지는 Auto Encoder이다. Stacked Auto Encoder의 구조는 가운데 히든 레이어를 기준으로 대칭인 구조를 가지며, 이러한 구조에 입력 노이즈를 추가하고 노이즈가 없는 원본 입력을 재구성하도록 학습시키는 과정을 거침으로써 Denoise를 진행한다.

## QRAFT Deep Learning 모델 구조

[Figure 6] AI Engine Procedure



자료: QRAFT Technologies

[Figure 6]은 당사의 딥러닝 모델의 가장 큰 특징을 보여주는 그림이다. 당사의 모델은 데이터를 호출하고, 모델이 학습하기 위한 형태의 전처리 과정과 딥러닝 학습에 중요한 부분인 하이퍼파라미터 튜닝, 마지막으로 최적 모델 포트폴리오까지 전 과정이 End to End로 한번에 이루어진다. 이는 자산배분을 위한 문제정의와 데이터만 주어진다면, 자신이 원하는 유니버스 내에서 얼마든지 원하는 모델 포트폴리오를 비교적 쉽게 얻을 수 있음을 의미한다. 아래부터는 위의 각 단계별로 핵심적인 내용을 설명한다.

### 1. Data Set

엔진에 들어가는 데이터는 총 3가지가 존재한다. 실제 포트폴리오 구성 및 성과를 측정하는 Asset Index Data, Regime Detection을 위한 Macro Economics Data 그리고 Valuation Data를 사용한다. 당사는 이러한 데이터 소스를 자체 개발한 Kirin API를 통해 손쉽게 접근하여 높은 유연성을 바탕으로 활용 중에 있다.

(1) **Asset Class Index** : 데이터는 DataStream과 Compustat을 활용한다. 각 자산들의 개별 종목 데이터를 사용하는 것이 아닌 각 자산의 Index 데이터를 사용한다. 그 이유는 Index의 경우 데이터 길이가 길며, 추종하는 ETF가 존재해 포트폴리오 구성이 가능하기 때문이다.

(2) **Macro Economics Data** : FRED 데이터를 활용하며, Credit Spread, Inflation, WTI Oil Price, GDP, M1, M2, Effective Federal Funds Rate 등을 호출하여 사용한다.

(3) **Valuation Data**: Compustat 데이터를 활용하며, S&P 500 P/E, Dividend Yield 등 현재 시장 Valuation을 활용 할 수 있는 데이터들을 호출하여 사용한다.

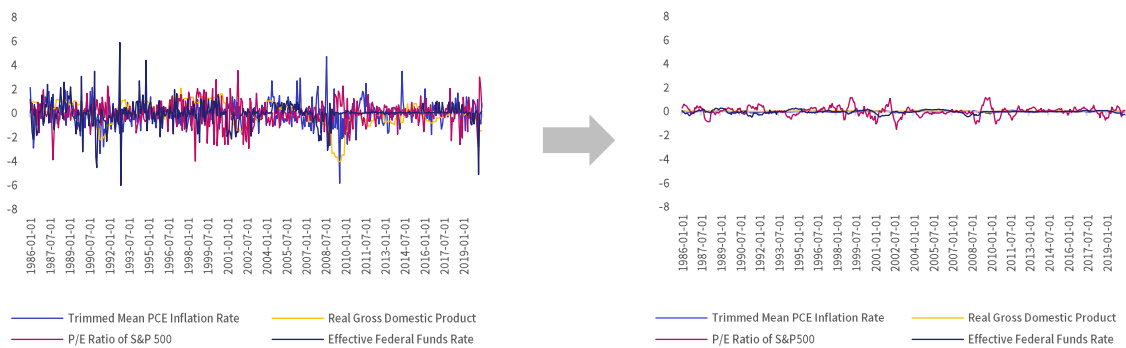
3가지 타입의 데이터들은 사전에 레이블링 된 최적의 포트폴리오를 목적함수로 두고 딥러닝 알고리즘을 통해 최적화 과정을 거친다. 이러한 과정을 통해 최적화된 모델 포트폴리오를 구성함으로써 낮은 위험 대비 높은 수익을 이끌어내는 것을 목표로한다.

## 2. Pre-Processing

Input 데이터의 노이즈로 인한 정상성 문제와 충분한 표본의 부족과 같은 문제를 해소하기 위해 앞서 언급한 바와 같이 노이즈를 제거한 후에 ADF-Test를 통한 정상성 테스트와 데이터 Imputation 과정을 거친다.

### (1) Denoise & Stationarity Check

[Figure 7] MA Denoise



자료: QRAFT Technologies, FRED

위의 [Figure 7]은 노이즈 제거 방법 중 가장 간단한 MA Denoise 방법을 통한 예시이다. 차분한 시계열 데이터의 노이즈를 제거하여 정보를 추정하게 된다.

[Figure 8] Stationary Check (ADF-Test)

```

=====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
=====
    
```

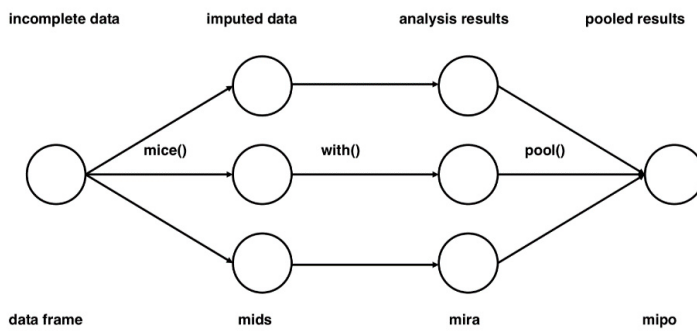
Source: QRAFT Technologies

당사의 전 처리(Pre-Processing) 모듈은 [Figure 8]과 같이 ADF Test를 통해 단위근(Unit-root) 검정을 자동으로 수행하고, 필요에 따라 전 처리를 진행함으로써 Input 데이터의 정상성을 유지하는 작업을 시행한다. 모델의 실행가능성을 점검하기 위해 Input 데이터의 정상성을 검증하는 것을 필수적인 과정이다. 이러한 과정을 통해, 정상적이지 않은 Input 데이터를 바탕으로 생성된 모델, 특히나 파라미터 수가 다소 존재하는 학습과정의 특성으로 인해 과거 특정 구간에 오버 피팅 될 가능성을 크게 줄일 수 있다.

## (2) Imputation

우리가 갖고 있는 금융데이터의 경우 무작위적으로, 혹은 데이터 수집 시기의 한계 등 다양한 원인으로 인해 결측 값을 갖게 되는 경우가 많다. 이는 데이터 분석에 많은 제약을 부여하게 되는데, 당사는 현재 가용 가능한 데이터와의 관계를 통해 결측 데이터를 가늠하여 사용하는 방법을 활용한다.

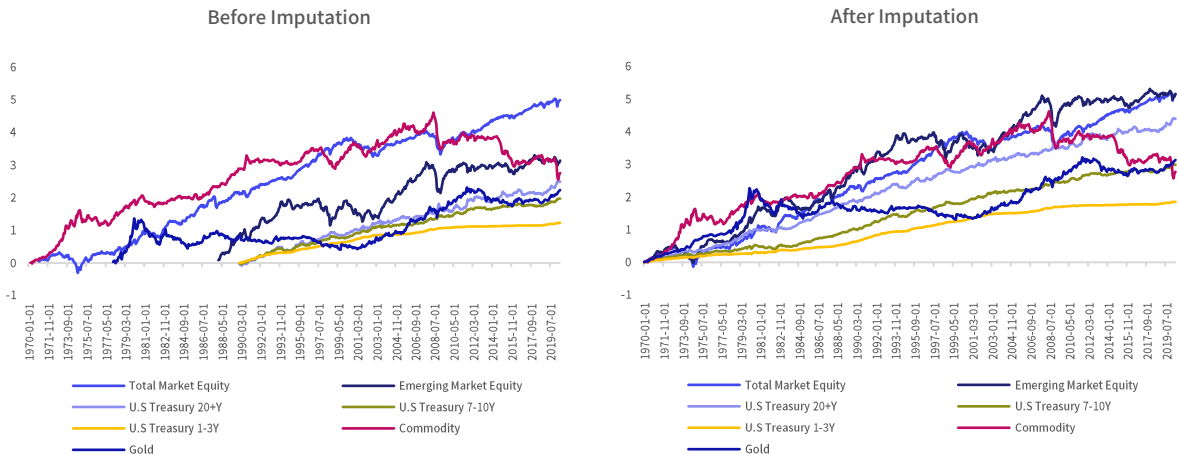
[Figure 9] Main Steps used in Multiple Imputation



자료: MICE : Multiple Imputation by Chained Equations in R

[Figure 9]은 Multiple Imputation by Chained Equation(이하 MICE)의 모식도를 나타낸다. 과거 결측 데이터는 MICE imputation 방법을 활용해 대체데이터로 추정한다. 이는 시계열 데이터 뿐만 아니라, 회계분류로 인한 데이터 Missing 조정, Outlier 데이터 조정 등에도 활용할 수 있다. 아래 [Figure 10]은 MICE를 통한 데이터 Imputation의 예시이다.

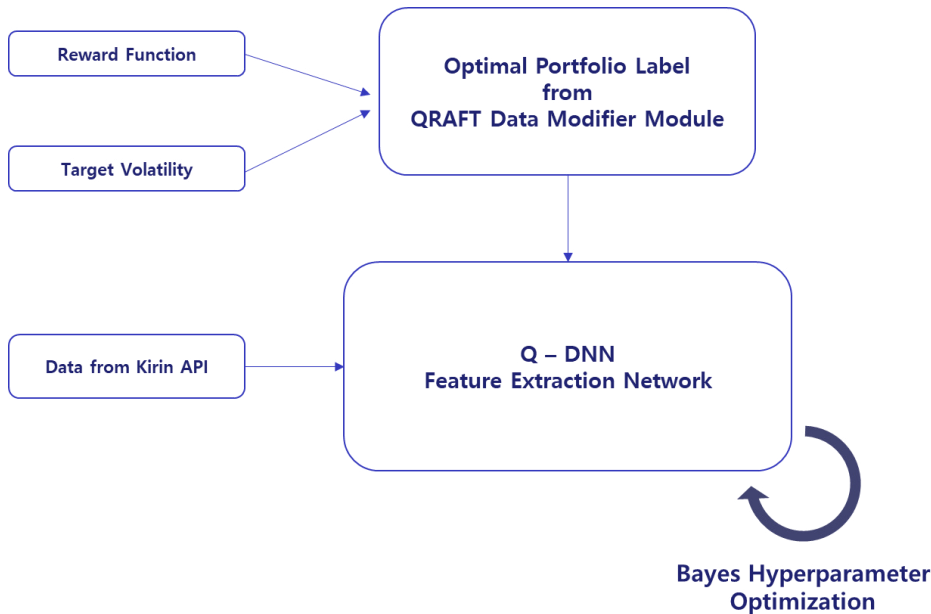
[Figure 10] Before & After Imputation



자료: QRAFT Technologies, DataStream

### 3. Model Structure

[Figure 11] Structure of Model



자료: QRAFT Technologies

[Figure 11]은 당사에서 사용하고 있는 Deep Learning 엔진을 모식도를 나타낸다. 어떠한 현실 문제를 딥러닝을 이용하여 해결하려면 3가지가 필요하다. 가장 먼저 어떤 것을 학습하려는지에 대한 문제 정의와 그 목표를 학습할 수 있게 하는 데이터, 마지막으로는 그 문제를 잘 학습할 수 있게 하는 모델이다.

가장 먼저 자산배분의 문제를 딥러닝을 적용할 때, 단순히 수익률을 예측하거나 움직임을 예측하려는 것은 적절하지 않은 상황을 초래할 수 있다. 주식, 채권, 원자재 등 개별 자산군들은 그 특성이 굉장히 다를 뿐만 아니라 거시경제 지표 및 국면에 따라서도 영향을 받는 데이터가 다르고, 각각의 상호작용에 의해 포트폴리오 전체에 미치는 영향이 예상한 것과는 전혀 다를 수 있기 때문이다. 따라서 당사의 모델에서는 이러한 요인을 간접적으로 반영해주기 위해, 자산군들의 수익률을 예측하는 것이 아니라 최적 포트폴리오를 달성할 수 있는 비중을 학습하고 예측하는 것을 목표로 한다. 즉, 미래의 수익률을 알고 있다는 가정하에 구성할 수 있는 최적 포트폴리오 비중을 간접적으로 학습시켜 데이터 내에서 최적의 행동을 가져가는 방식을 택하고 있다. 이를 위해 투자자의 목적을 반영할 수 있는 포트폴리오 성과지표(Reward)를 설정하고, 이때 투자자의 리스크 감내 수준을 반영한 Target Volatility를 설정한다. 또한 미래 행동의 불확실성을 반영할 수 있는 할인율을 적용하고 학습할 미래시점의 최적 포트폴리오 비중을 생성하게 된다. 당사의 자산배분 모델은 이렇게 생성된 비중을 딥러닝이 학습할 라벨(Optimal Portfolio Label)로 사용하게 된다.

최적의 행동 레이블을 생성한 후에는 데이터의 특성들을 잘 추출해 낼 수 있는 모델의 구조를 구성하는 단계가 필요하다. 특정 시점에서 최적의 포트폴리오를 구성하기 위해서는 당시 상황을 나타내는 Macro 데이터와 Valuation 데이터가 필요한데, 이 부분은 당사의 Kirin API를 사용한다. Kirin API를 통해 미래 참조를 해소하며 과거 각 시점에서 실제로 활용할 수 있었던 데이터만을 이용하여 딥러닝 학습을 진행한다.

딥러닝 모델 구조의 경우, 본 페이지에서 세부적으로 설명하긴 어려우나 핵심적인 내용을 언급하자면 다음과 같다. 복잡한 비선형 관계를 가진 금융데이터를 학습하기 위해서 당사에서는 시계열의 특징을 잘 추출해 낼 수 있는 Recurrent 계열의 신경망 구조와 정적인 데이터에 적합한 Fully connected 및 convolution 계열의 신경망 구조를 적절히 조합한다. 특히 변수들의 Cross sectional 사이에 존재하는 정보와 Time Series에서 존재하는 정보를 한 번에 고려해 추출할 수 있는 당사의 TS-CS module<sup>7</sup>를 사용한다. 또한 Noisy한 데이터에 강건하게 특징을 학습할 수 있는 Adversarial Training을 통해 안정적인 특징의 추출을 유도한다.

<sup>7</sup> 데이터의 CS와 TS특성을 딥러닝에 반영하기 위한 당사가 직접 개발한 딥러닝 구조이다.

## 4. Hyper Parameter Tuning

하이퍼파라미터란 학습을 수행하기 전에 설정하는 값을 의미한다. 가령 학습율(Learning Rate), 미니배치 크기(Minibatch size), L2 정규화 계수(L2 Regularization coefficient) 등이 있다. 당사는 당사의 딥러닝 엔진에 활용되어 지는 하이퍼파라미터 값들을 튜닝하기 위해 몇가지 우선순위를 두고 진행하고 있다.

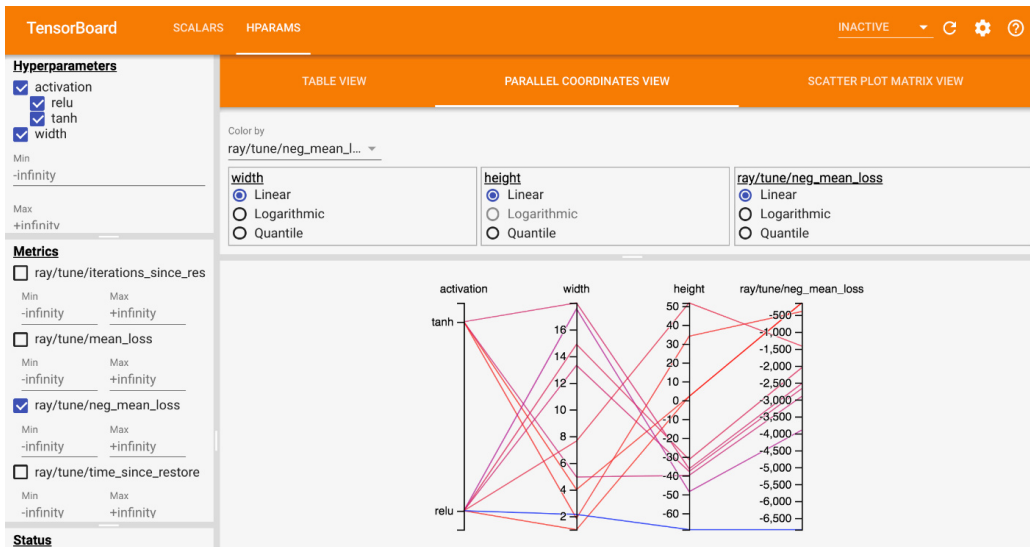
(1순위) Network Capacity: Neural Network Model의 노드 수와 레이어 수와 같은 환경 설정에 의해 정의된다. 너무 적은 Network Capacity는 Training Data Set를 적절히 학습하지 못해 Under Fitting되고, 너무 높은 Network Capacity는 Training Data Set을 너무 학습해 버려 Over Fitting이 된다.

(2순위) Learning Rate & Batch Size: Learning Rate는 한 번 학습할 때 얼마만큼 학습해야 하는지 학습 양을 의미하며 너무 크면 큰 값을 반환하고, 너무 작으면 갱신되지 않고 학습이 끝나버린다. Batch Size는 네트워크가 전체 데이터를 학습하는 한번의 epoch 동안 데이터를 나누어 주는 Iteration을 거치는데, 각 Iteration 마다 주는 데이터 사이즈를 의미한다. 적정 Learning Rate와 Batch-Size는 Network Capacity, Loss Function 등에 따라 달라지는데 이를 고려한 학습 환경을 구축해야 한다.

(3순위) Target Portfolio 하이퍼파라미터: 위 타겟이 될 포트폴리오 레이블을 구성하는 하이퍼파라미터 들을 잘 설정할 필요가 있다. 최적 포트폴리오 구성 시 고려되어야 할 하이퍼파라미터는 Lambda 값, Window Size, Objective Function(Sharpe Ratio, Sortino Ratio) 등이 있다.

(4순위) Early Stopping: 너무 많은 Epoch 값은 Over Fitting을 일으키고, 너무 적은 Epoch 값은 Under Fitting을 일으킨다. 이런 경우 Early Stopping은 많은 Epoch를 시행한 후, 특정 시점에서 멈추는 것을 의미한다.

[Figure 12] Hyperparameter Tuning with Ray Tune



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

하이퍼파라미터를 튜닝하는 과정은 크게 2단계로 나뉜다. 1단계는 직접 손으로 튜닝해보면서 Tensor Board 등으로 Learning Curve를 직접 체크하는 것이다. 가령 Batch Size가 커지면 적정 Learning Rate를 조정해주고, 위 언급한 4순위 Early Stopping에서 단계적인 Under Fitting 등이 발생하는지를 관찰한다. 2단계로는 Auto-ML config를 작성하고 NNI 혹은 Ray Tune을 이용한다. 얼마나 모델을 잘 Tuning 하는가에 따라 모델의 성능은 크게 상이하다. 매번 Tuning을 하기에는 현실적으로 불가능 하기에 Ray Tune과 Auto-ML 알고리즘을 잘 활용하는 것이 중요하다.

## 실증 분석

간단한 예시를 통해 딥러닝 모델의 성과를 비교하였다. 모델의 우수성을 확인해보기 위해, 자산배분에 많이 사용하는 방법들을 함께 살펴본다. 우선 전통적인 방법인 60/40 포트폴리오를 벤치마크로 두고 이와 함께 Risk Parity 포트폴리오(이하 RP), 평균-분산 포트폴리오(이하 MV), 최소 변동성 포트폴리오 (이하 Min-Vol)를 살펴본다. 다음으로는 당사의 Deep Learning모델(이하 DL)과 함께, 동일한 학습 구조에 모델만 변경하여 다양한 Machine Learning(이하 ML)기법을 적용한 포트폴리오 결과를 살펴본다. 이를 통해 당사의 DL이 큰 흐름을 함께하는 ML방법에 비해 우수성을 확인하고자 한다. 본고에서 비교를 위한 ML방법은 Support Vector Regression(이하 SVR), Decision Tree(이하 DT), Linear Regression(이하 LR), Gradient Boosting Regression(이하 GBR), Random Forest(이하 RF), Ridge, Lasso를 사용한다.

[Table 2] Application Data Set

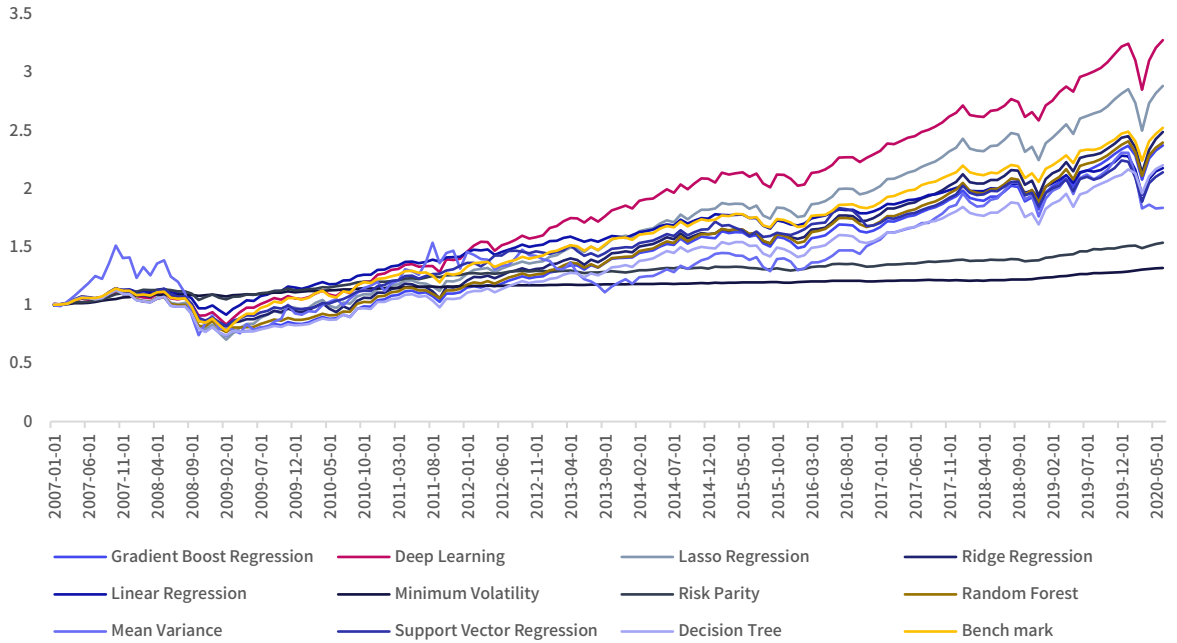
Category	Ticker	Name	Description
<b>Panel A : 7 Asset Class</b>			
Equity	WIL5TMK	Wilshire 5000 Total Market	Equity
Equity	MSEMKF\$	MSCI Emerging Markets USD	Emerging Country Equity
Bond	SPBDUSL	S&P US Treasury 20+ years	Treasury 20+Y
Bond	SPBDU10	S&P US Treasury 7-10 years	Treasury 7-10Y
Bond	SPBDUS3	S&P US Treasury 1-3 years	Treasury 1-3Y
Commodity	GSCITOT	S&P GS Commodity Index	Commodity
Gold	GSGCTOT	S&P GS Commodity Index - Gold	Gold
<b>Panel B : 11 Macro &amp; Valuation Class</b>			
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

자료: QRAFT Technologies, Compustat, DataStream, FRED

[Table 2]는 예제 분석을 위해 활용되어지는 데이터들을 나타낸다. 자산군 데이터의 종류는 크게 4가지로 주식, 채권, 원자재 그리고 금으로 구성되어 있다. 또한, 9개의 매크로 데이터와 2개의 Valuation 데이터를 활용한다. 분석에서의 In-sample 기간은 1970년 1월부터 2006년 12월까지이며 Out-of-sample 기간은 2007년 1월부터 2020년 6월이다. In-sample에서 추정된 각 모델의 파라미터를 이용하여, Out-of-sample에서 개별 자산의 weight를 산출하고 포트폴리오를 구성한다. 이때 파라미터 최적화를 통해 In-sample 기간 중 최근 10%의 기간을 검증용 데이터로 활용한다. 훈련을 위해 사용된 레이블은 미래 12개월동안 Sortino Ratio를 최대화하는 함수를 사용하였으며, 포트폴리오 변동성의 타겟은 연 6%에서 17% 사이로 하였다. 투자자의 입장에서 상방 변동성의 경우에는 이익의 형태로 돌아오기에 하방 위험만 고려하는 것이 투자 자체의 위험 대비 수익률을 더 잘 나타낸다고 볼 수 있다. 따라서 Sortino Ratio를 최대화하는 함수를 사용하였다.



[Figure 13] Cumulative Return of Portfolios



자료: QRAFT Technologies, Compustat, DataStream, FRED

머신 러닝의 모델들은 벤치마크인 60/40 포트폴리오의 성과를 대부분 하회하고 있으며, Lasso 모델과 DL 모델이 아웃 퍼폼을 하고 있음을 보여준다. Lasso과 DL을 비교해보면, Lasso은 BM 대비 성과가 일정하지 않은 모습을 보여주는데, 딥러닝 모델은 BM 대비 꾸준한 초과수익을 보이고 있다.

[Table 3] 각 모델 포트폴리오 성과지표

이 표는 표본 기간 동안의 각 포트폴리오의 성과를 나타낸다. 아래의 표의 값은 연 단위 값이며, Risk Free의 경우, 미국 국채 3년물을 사용하였다. 또한 Bench-mark는 US주식 45%, 신흥국 주식 15%, 미국 국채 10년물 40%로 설정하였다.

	Mean	Vol	Sharpe	Sortino	MDD
<b>Panel A: Traditional Approach</b>					
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
<b>Panel B: Machine Learning Approach</b>					
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

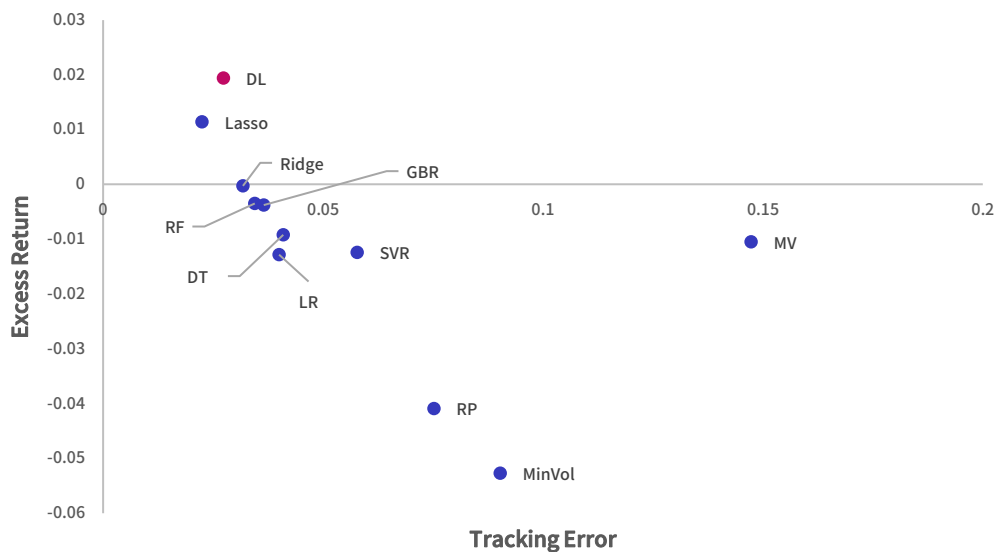
자료: QRAFT Technologies, Compustat, DataStream, FRED

[Table 3]는 OOS구간 백테스트 결과를 나타낸다. Panel A는 벤치마크 포트폴리오를 비롯해 전통적인 방법인 RP, MV 그리고 Min-Vol의 포트폴리오 결과이다. Panel B는 자사의 딥러닝 모델의 우수성을 확인하기 위해, 동일한 학습 구조에 다양한 ML 모델을 적용한 결과를 나타낸다.

먼저 Panel A의 전통적 접근 방법들은 모두 평균수익률이 벤치마크 대비 하회하는 결과를 보여준다. 반면 MV의 경우 위험 조정 성과 지표인 Sharpe Ratio와 Sortino Ratio가 벤치마크 대비 낮은 모습을 보여주고 있지만, Min-Vol과 RP의 경우 상회하는 모습을 보여준다. 하지만 이러한 성과의 원인은 낮은 수익률과 변동성에 기인한다. 따라서 두 방법 모두 수익률 관점에서는 벤치마크 대비 우월하다고 보기는 어렵다.

다음으로 Panel B를 살펴보면, Lasso와 DL 방법을 제외한 다른 방법들은 벤치마크 대비 수익률과 위험조정 성과가 좋지 않다. Lasso의 경우, 연평균 수익률이 7.61%로 벤치마크의 연평균수익률 6.47%를 아웃 퍼폼 하는 모습을 보여준다. 하지만 위험 조정 지표인 Sharpe Ratio와 Sortino Ratio를 살펴보면 벤치마크를 하회하는 모습을 보여, 모델의 성능이 좋다고 하기 어렵다. 반면 당사의 DL의 경우, 연평균수익률이 8.40%로 벤치마크 평균수익률을 크게 상회한다. 또한 Sharpe Ratio와 Sortino Ratio도 가장 높게 나타나기에 다른 ML 방법에 비해 우수하다고 할 수 있다.

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



자료: QRAFT Technologies, Compustat, DataStream, FRED

[Figure 14]은 초과 수익률과 추적오차를 나타낸 그래프이다. 벤치마크와 추적오차가 발생하여도, DL과 Lasso만이 초과 수익을 얻고 있음을 알 수 있다. 나머지 전통적인 포트폴리오 구성 방법과 ML을 적용한 포트폴리오는 초과수익을 내지 못하는 모습을 보인다.

**[Table 4] 각 모델 포트폴리오의 Information Ratio**

	Mean return	Excess return	Tracking Error	Information Ratio
<b>Panel A: Traditional Approach</b>				
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
<b>Panel B: Machine Learning Approach</b>				
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

자료: QRAFT Technologies, Compustat, DataStream, FRED

[Table 4]는 각 포트폴리오의 정보비율<sup>8</sup>을 산출한 결과이다. 벤치마크 대비 초과수익률을 벤치마크와의 추적 오차로 나눈 값을 정보비율이라고 하며, 포트폴리오가 얼마나 일관성 있게 벤치마크를 초과하는 성과를 내는지를 측정하고자 하는 척도이다. DL과 Lasso를 제외한 모든 방법의 정보비율은 음의 값을 나타내고 있다. 반면 DL의 정보비율은 0.7079로 Lasso의 정보비율 0.5089에 비해 크게 나타난다. 이는 IR로 각 전략을 평가하더라도, DL의 방법이 우수하다는 것을 보여준다.

**[Table 5] 각 모델 포트폴리오의 초과 수익 검증**

이 표는 표본 기간 동안의 각 포트폴리오의 벤치마크 대비 초과수익률을 나타낸다. 아래의 표의 값은 월 단위 값이며, 지연 시차 12의 Newey and West(1987) t-통계량을 사용하였다.

	Coefficient	Std. Error	t-value	p-value
<b>Panel A: Traditional Approach</b>				
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
<b>Panel B: Machine Learning Approach</b>				
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

자료: QRAFT Technologies, Compustat, DataStream, FRED

<sup>8</sup> Bacon, 2008, Practical Portfolio Performance Measurement & Attribution, Wiley

또한 각 모델이 창출한 초과수익률의 크기가 통계적으로 유의한 지 살펴볼 필요가 있다. 이를 위해 벤치마크 대비 초과수익률의 Newey and West(1987) T-test로 유의성을 확인한다. [Table 5]의 Panel A를 살펴보면, 전통적인 방법에서는 모든 방법이 음의 초과 수익을 보이며, 특히 MV와 RP의 경우 통계적으로 유의한 음의 값을 보인다.

Panel B에서는 Lasso와 DL을 제외한 모든 방법의 초과 성과가 통계적으로 유의하지 않음을 보여준다. 이는 포트폴리오 수익률이 벤치마크를 일시적으로 상회하거나 하회하더라도 그 유의성이 뚜렷하게 나타나지 않음을 의미한다. 반면 DL로 구성된 포트폴리오의 초과 수익률은 통계적으로 유의한 양의 값을 가지며, 그 크기는 Lasso에 비해서 큰 것을 알 수 있다. 이러한 결과는 [Table 4]에서의 성과 경향과 일치하는 모습이며, DL 포트폴리오 성과의 우수성을 보여준다.

투자 전략의 성과가 측정되는 구간이 길어지게 되면, 전체 구간 동안 측정된 투자 성과는 일부 특정 시기에 의해 왜곡될 가능성이 존재한다. 따라서 각 시점마다 특정기간 동안 투자하여 얻을 수 있는 수익률(Rolling Return)의 분포를 살펴보는 것이 중요하다. 본 페이지에서는 각 포트폴리오의 Rolling Return이 벤치마크 대비 얼마만큼의 확률로 아웃 퍼폼 하였는지를 살펴본다.

[Table 6] 각 모델들의 Rolling Return Win ratio

	3-month	6-month	12-month	24-month
<b>Panel A: Traditional Approach</b>				
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
<b>Panel B: Machine Learning Approach</b>				
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

자료: QRAFT Technologies, Compustat, DataStream, FRED

[Table 6]는 벤치마크 대비 포트폴리오의 Rolling Return을 산출하여, Win ratio를 산출한 값이다. DL모델의 경우, 3개월 Rolling Return의 승률 60.63%에서 24개월 Rolling Return의 승률이 88.97%로 투자기간이 장기화될수록 승률이 더욱 높아지는 경향을 보인다. Lasso 모델의 경우 3개월 Rolling Return의 승률은 66.88%로 DL모델보다 높은 값을 보이는데, 투자기간이 장기화 될수록 승률의 상승폭은 DL에 비해 낮은 모습을 보인다. 이를 통해 투자기간이 길어질수록 DL 모델이 벤치마크 대비 초과수익을 낼 수 있는 확률이 유의하게 상승함으로써 DL 모델의 우수성을 확인할 수 있다.

## Conclusion

자산배분 전략은 다양한 투자 주체와 다양한 시장에서 매우 적극적으로 활용되어 지고 있다. 자산배분 전략은 맹목적으로 높은 수익률을 추구하기 보다는 위험도 함께 고려하는 투자의 안정성이 중요하다. 최근에는, 단순 60/40 포트폴리오와 같은 전통적 자산배분에서 벗어나 Mean-Variance, Risk Parity 전략 등이 많이 활용되어 지고 있다. 하지만, 이러한 전략들은 단순 시계열의 연속성을 가정하기 때문에 불 연속적인 금융시장에서의 투자전략으로서 활용도의 한계점이 분명하다. 이러한 한계점을 해소하고 효과적인 자산배분을 위해 당사는 Deep Learning을 활용한 방법을 제시한다.

본 페이지는 당사가 실제 활용 중인 Deep Learning Asset Allocation Model의 간략한 모델 구조를 설명하고 Deep Learning을 통한 자산배분 전략의 당위성을 제시하고자 하였다. 이를 위해 진행한 실증분석에서는 전통적 자산배분 전략인 60/40 Portfolio를 벤치마크로 두어 성과를 점검하였다. 또한, 비교적 잘 알려진 Machine Learning Model 과의 성능을 비교해 봄으로써, Deep Learning 모델의 우월함을 다시 한번 확인하였다. 분석 결과를 살펴보면 당사의 Deep Learning Asset Allocation Model의 성과가 가장 우수하였다.

하지만, 이러한 성과에 불구하고 다른 Machine Learning 방법론들이 좋은 방법이 아니라고 말하기 힘든 점은 존재한다. 그 이유는 Machine Learning과 Deep Learning은 학습 데이터에 따라 향후 모델의 성과가 크게 좌우되기 때문이다. 이러한 점을 고려한 모델의 성능을 살펴보기 위해서는, 시간이 지남에 따라 지속적으로 재학습을 진행하고, 재학습된 모델의 결과를 다시 사용하는 Rolling-Window 방식이 고려되어야 한다. 본 페이지에서는 위의 과정이 생략되었으나, 당사가 실제로 사용 중에 있는 자산배분 모델은 위의 과정을 포함하여 모델을 지속적으로 업데이트 한다. 이러한 과정을 통해 더욱 현실적이고 좋은 성능을 가진 모델을 적용할 수 있다. 이러한 분석 과정에 대해서는 추후 발간될 AI Report에서 하나의 큰 골자로 잡아 발간할 예정이다.

금융 데이터를 활용하여 투자 전략을 수립하고 집행하는데 있어 무수히 많은 어려움을 직면한다. 다양한 문제점들을 당사는 다양하고 효과적인 인공지능 기술로 풀어가고 있다. 자산배분에서 Deep Learning의 활용은 완전 새로운 영역이 아니라 그간 전통적 투자 방법의 확장선이라는 해석이 더욱 정확하다. 전통적인 자산배분 전략들은 금융 시계열의 연속성만을 가정하며, All Weather 전략은 Naïve 하게 시장 국면이 발생할 확률을 동일하게 가져간다는 차이가 있지만 모두 과거 금융 데이터를 활용한다는 점에서는 큰 골자 안에 함께 있다. 이러한 큰 골자 안에서 Deep Learning은 새로운 방향성의 제시일 뿐, 세상에 없던 투자 전략을 만들어내는 것이 아니라는 점에서 그 가능성은 무한하다.

### Disclaimer

- \* 본 자료는 정보제공을 위해 작성되었으며, 펀드 등 금융투자상품의 판매를 권유하기 위한 목적으로 사용될 수 없습니다.
- \* 본 자료에 수록된 내용은 당사의 퀀트 리서치 팀의 추정치로써 오차가 발생할 수 있으며, 그 정확성이나 완전성을 보장할 수 없으므로, 본 자료와 관련된 투자의 최종 결정은 투자자 자신의 판단으로 하시기 바랍니다. 따라서 본 자료는 어떠한 경우에도 투자 결과에 대한 법적 책임소재의 증빙자료로 사용될 수 없습니다.
- \* 본 자료는 크래프트 테크놀로지스가 특허출원 중이거나 특허 등록한 내용을 포함하고 있습니다. 본 자료는 당사의 허락 없이 무단 복제 및 배포할 수 없습니다.
- \* 크래프트 테크놀로지스는 AI 기술을 활용해 자산운용업의 비효율성을 혁신하고 있는 기업입니다. 데이터처리부터 알파리서치, 그리고 포트폴리오 주문집행까지 자산운용업의 각 단계별 비효율성을 기술로써 해결하고, 이를 통해 높은 수준의 알파를 낮은 비용으로 제공하는 것을 목표로 합니다.