KIPS Trans. Comp. and Comm. Sys.
Vol.8, No.1 pp.17~28 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2019.8.1.17

Blockchain Based Financial Portfolio Management Using A3C

Ju-Bong Kim[†] · Joo-Seong Heo[††] · Hyun-Kyo Lim[†††] · Do-Hyung Kwon[††††] · Youn-Hee Han[†††††]

ABSTRACT

In the financial investment management strategy, the distributed investment selecting and combining various financial assets is called portfolio management theory. In recent years, the blockchain based financial assets, such as cryptocurrencies, have been traded on several well-known exchanges, and an efficient portfolio management approach is required in order for investors to steadily raise their return on investment in cryptocurrencies. On the other hand, deep learning has shown remarkable results in various fields, and research on application of deep reinforcement learning algorithm to portfolio management has begun. In this paper, we propose an efficient financial portfolio investment management method based on Asynchronous Advantage Actor-Critic (A3C), which is a representative asynchronous reinforcement learning algorithm. In addition, since the conventional cross-entropy function can not be applied to portfolio management, we propose a proper method where the existing cross-entropy is modified to fit the portfolio investment method. Finally, we compare the proposed A3C model with the existing reinforcement learning based cryptography portfolio investment algorithm, and prove that the performance of the proposed A3C model is better than the existing one.

Keywords : Reinforcement Learning, Financial Portfolio Management, A3C, Cryptocurrency, Investment Engineering

A3C를 활용한 블록체인 기반 금융 자산 포트폴리오 관리

김 주 봉[†] · 허 주 성[††] · 임 현 교[†††] · 권 도 형[††††] · 한 연 희[†††††]

요 약

금융투자 관리 전략 중에서 여러 금융 상품을 선택하고 조합하여 분산 투자하는 것을 포트폴리오 관리 이론이라 부른다. 최근, 블록체인 기반 금융 자산, 즉 암호화폐들이 몇몇 유명 거래소에 상장되어 거래가 되고 있으며, 암호화폐 투자자들이 암호화폐에 대한 투자 수익을 안정적으로 올리기 위하여 효율적인 포트폴리오 관리 방안이 요구되고 있다. 한편 딥러닝이 여러 분야에서 괄목할만한 성과를 보이면서 심층 강화학습 알고리즘을 포트폴리오 관리에 적용하는 연구가 시작되었다. 본 논문은 기존에 발표된 심층강화학습 기반 금융 포트폴리오 투자 전략을 바탕으로 대표적인 비동기 심층 강화학습 알고리즘인 Asynchronous Advantage Actor-Critic (A3C)를 적용한 효율적인 금융 포트폴리오 투자 관리 기법을 제안한다. 또한, A3C를 포트폴리오 투자 관리에 접목시키는 과정에서 기존의 Cross-Entropy 함수를 그대로 적용할 수 없기 때문에 포트폴리오 투자 방식에 적합하게 기존의 Cross-Entropy를 변형하여 그 해법을 제시한다. 마지막으로 기존에 발표된 강화학습 기반 암호화폐 포트폴리오 투자 알고리즘과의 비교평가를 수행하여, 본 논문에서 제시하는 Deterministic Policy Gradient based A3C 모델의 성능이 우수하다는 것을 입증하였다.

키워드 : 강화학습, 금융 포트폴리오 관리, A3C, 암호화폐, 투자공학

1. 서 론

금융 투자 이론 중 하나인 포트폴리오 투자 이론은 자산을 분산투자하여 포트폴리오를 만들어 기존의 투자 전략보다 위험을 감소시키는 것을 말한다. 한편 블록체인 기술을 기반으로 만들어진 암호화폐인 비트코인은 2009년 이후 급격한 성장세를 보이며 발전해온 투자자산의 한 종류이다[1]. 비트코인은 중앙은행 없이 세계 어느 곳에서나 P2P방식으로 금융거래를 가능하게 했고, 비트코인에 이어 수많은 알트코인들이 생겨나며 블록체인 기반 암호화폐 생태계를 이루고 세계 곳곳의 거래소에서 거래되고 있다. 포트폴리오 투자 이론은 시세 변화가 민감한 암호화폐 시장에서도 안정적으로 투자를 할 수 있도록 만들어 준다. 하지만 암호화폐 시장은 주식 시장과는 다르게 24시간 개장이 되어 있기 때문에, 투자자가 직접 포트폴리오 투자 관리를 끊임없이 지속하는 것은 매우 어

※ 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2016R1D1A3B03933355).
※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1A6A1A03025526).
† 준 회 원 : 한국기술교육대학교 컴퓨터공학부 석사과정
†† 준 회 원 : 한국기술교육대학교 컴퓨터공학부 석사수료
††† 준 회 원 : 한국기술교육대학교 창의융합공학협동과정 ICT융합 박사과정
†††† 준 회 원 : 한국기술교육대학교 창의융합공학협동과정 ICT융합 석사과정
††††† 종신회원 : 한국기술교육대학교 컴퓨터공학부 교수
Manuscript Received : December 4, 2018
Accepted : December 13, 2018
* Corresponding Author : Youn-Hee Han(yhhan@koreatech.ac.kr)

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/ licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

려운 일이다. 최근에, 그 해결책으로 주식투자 전략을 알고리즘화한 봇을 사용하고자 하는 노력이 많이 이루어지고 있다. 그 예들로 2016년 12월에 공개된 Gunbot [2]과 2017년 9월에 공개된 Profit Trailer Bot [3]이 있다. 하지만, 이러한 봇들은 투자자가 직접 개별적인 매수/매도 전략을 지정해야하기 때문에, 그러한 매수/매도 전략에 대한 지식이 없으면 봇의 운영이 쉽지 않다.

최근 딥러닝을 포함한 머신 러닝이 여러 분야에서 괄목할 만한 성과를 보이면서 머신러닝으로 암호화폐 거래를 시도하는 연구가 많이 진행되어 왔다. 하지만 대부분의 연구는 정량적인 주식 가격의 흐름을 예측하는 것들이 대부분이며 알고리즘 내부에서도 주식이론을 많이 도입하고 있다[4, 5]. 또한 제안하는 알고리즘의 성능은 예측 정확도에만 의존하며, 그러한 예측 정확도에 입각하여 거래 시그널을 생성하는 방향으로 연구가 이루어지고 있다[6, 7].

최근 강화학습은 뉴럴 네트워크와 결합이 되어 높은 성능을 보인다. 대표적인 강화학습 알고리즘 예로 벨만 최적 방정식에서 비롯된 DQN (Deep Q-Network)이 존재하며, 이는 탐욕적으로 오프폴리시(Off-Policy) 시간차 제어 방식의 학습을 통하여 비교적 높은 성능의 모델을 생성한다. 하지만, 과거의 행동(Action)을 충분히 메모리에 저장을 한 이후 학습에 사용하기 때문에, 에피소드가 매우 길거나 끝이 없는 환경(Environment)에서는 DQN을 활용하기 어렵다. 따라서 각 행위 스텝별로 학습이 가능하면서도 정책(Policy) 네트워크와 가치(Value) 네트워크를 분리하여 사용하는 A2C (Advantage Actor-Critic), A3C (Asynchronous Advantage Actor-Critic)와 같은 강화학습 알고리즘들이 최근 발표되고 그 성능과 효율이 주로 컴퓨터 게임 분야에서 입증된 바 있다 [8, 9, 10].

그리고 최근 암호화폐 포트폴리오 관리 전략에 활용되는 시도가 학계에서 발표되고 있다. 강화학습 에이전트가 대신 전략을 고안 하도록 하여 투자의 경험이 없거나 투자지식이 적은 사람도 안정적인 자산 관리를 할 수 있도록 만드는 것이 목표이다. 이중 가장 대표적인 연구[11]은 Deterministic Policy Gradient 알고리즘을 사용한 것이다. 이 논문에서는 Ensemble of Identical Independent Evaluators (EIIE) 뉴럴 네트워크 토폴로지와 Portfolio-Vector Memory (PVM) 등의 기법을 활용하여 투자 시뮬레이션 환경에서 백테스트(Back-Test)를 수행하였고 머신러닝을 활용한 포트폴리오 관리에 대한 기존 연구들보다 성과가 뛰어났다.

하지만, 한정된 시간 내에 투자 수익을 보다 높일 수 있는 강화학습 모델을 구성하기 위해서는 여러 강화학습 에이전트를 비동기적으로 활용하는 A3C 알고리즘 접목이 필요하다. 따라서 본 논문에서는 포트폴리오 투자 이론에 최신 강화학습 알고리즘인 A3C를 접목한 연구 수행 결과를 제시한다. 금융 포트폴리오 투자 이론에 A3C를 접목하려는 이유는 다음과 같다. 첫 번째는 오프폴리시 모델이 비동기 다중 에이전트를 통해 학습되기 때문에 하나의 에이전트가 직면할 수 있는 지역 최적화(Local Optimization) 문제에서 좀 더 자유롭기 때문이다[7, 8]. 두 번째는 에이전트의 투자 종목 선택 및 배분율 결정에 직접 관련된 정책 네트워크와 투자 수익 성능 평가에 관련된 가치 네트워크를 분리시켜서, DQN과 다르게 그러한 분리된 정책의 가치 판단을 통하여 최종 목적인 정책에 대한 학습을 보다 효율적으로 수행할 수 있기 때문이다[9].

한편, 암호화폐 포트폴리오 관리에 A3C를 접목시키는 과정에서 기존의 Cross-Entropy 함수를 그대로 적용할 수 없기 때문에 기존의 Cross-Entropy를 변형하여 정책 네트워크 업데이트에 사용하였다. 한편, 기존에 발표된 강화학습 기반 포트폴리오 투자 알고리즘과의 비교평가를 수행하여, 본 논문에서 제시하는 A3C 모델의 성능이 Deterministic Policy Gradient 모델보다 우수하다는 것을 입증하였다.

본 논문의 구성은 다음과 같다. 2장에서는 강화학습과 포트폴리오 투자 관리에 관한 기존 연구에 대해 설명하고, 3장에서는 데이터 처리에 관한 내용을 설명한다. 최신 강화학습 알고리즘인 A3C를 적용하여 만든 모델인 “Deterministic Policy Gradient based A3C”를 4장에서 제시한다. 이어 5장에서는 Deterministic Policy Gradient 모델과의 비교평가 결과 또한 보인다.

2. 관련 연구

2.1 강화 학습

강화학습의 목표는 에이전트가 환경과 상호작용하며 기대 누적 보상 $E[R]$을 최대한 높이는 것이다. 에이전트는 임의의 스텝 t에서 환경의 상태(State) s_t를 관찰하고, 파라미터 집합 θ를 지닌 모델의 정책 π_θ를 통해 도출한 행동(Action) a_t을 수행하고, 그 행동에 대한 보상(Reward) r_t를 받는다. 한편, 에이전트는 임의의 상태 t에서 시작하여 T 스텝에서 에피소드가 끝난다고 가정할 때, 보상 감쇠율(Discount Factor) γ ($\gamma \in (0,1]$)를 고려한 다음 Equation (1)과 같은 상태 t에서의 누적 보상을 얻는다[9, 10].

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{1}$$

Action-Value 함수 $Q^{\pi_\theta}(s_{t,}, a_t)$는 현재 상태 s_t를 관찰한 뒤, 현재 정책 π_θ에 따른 행동 a_t에 대한 누적보상 R_t의 기댓값이다. 최적의 Action-Value 함수에 대한 수식은 다음과 같다.

$$Q^*(s_t, a_t) = argmax_{\pi_\theta} Q^{\pi_\theta}(s_{t,}, a_t) \tag{2}$$

Equation (2)는 재귀적인 형태를 가지는 아래와 같은 벨만 최적방정식으로 표현이 가능하다.

$$Q^*(s_t, a_t) = E\left[r_{t+1} + \gamma max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})\right] \tag{3}$$

Equation (3)은 강화학습의 목적인 누적보상 R_t을 최대로 높이는 데 사용된다[12-14].

일반적으로 강화학습에서 시간차 방법에 의한 Equation (3)은 정책 평가 업데이트에 사용되며 탐욕적으로 Action-Value 함수를 개선하여 정책을 업데이트한다. 한편, 정책의 기울기(Gradient) 방향으로 정책을 직접 업데이트하는 방법이 존재한다. 구체적으로, 임의의 스텝 t에서의 정책 π^t을 기반으로 Action-Value 함수의 기울기 방향으로 업데이트 하여 새로운 스텝에서의 정책 π^{t+1}을 얻을 수 있다. 이와 같은 방법은 아래 수식으로 나타내며

$$\pi_\theta^{t+1} = \pi_\theta^t + \gamma E_{\pi_\theta^t}[\nabla Q^{\pi_\theta^t}(s_{t+1}, a_{t+1})] \quad (4)$$

Chain-Rule 에 입각하여 계산하면 아래와 같은 정책 업데이트 수식에 의해 정책의 기울기 방향으로 정책 파라미터들을 개선시킨다[15].

$$\pi_\theta^{t+1} = \pi_\theta^t + \gamma E_{\pi_\theta^t}[\nabla_{\pi^t} Q^*(s,a) \nabla_a Q^*(s,a)] \quad (5)$$

하지만 시간차 방법에 의해서 정책 파라미터들을 업데이트 시킨다면 업데이트가 수행된 직후 스텝에서는 개선된 정책 파라미터들이 쓰일 것이다. 이는 업데이트된 정책이 배포되는데 대한 명백한 평가가 이루어지기 전이므로 이에 대한 보완이 필요하다.

그래서 스텝이 진행 중일 때 정책 파라미터들을 업데이트 하지 않고 정책에 기반 하여 모든 상태 값들을 고려하면서 업데이트하는 방법이 DPG (Deterministic Policy Gradient) 방법이다. 이 방법에서는 정책 파라미터 θ에 대하여 행동 $a_t = \pi_\theta(s_t)$을 정의하며, 정책에 의한 행동의 보상 값을 정책의 기울기 방향으로 업데이트 시킨다. 수식으로는 다음과 같다[11, 15].

$$R(s_t, a_t) = E(\sum_t^T \gamma^t r_t | \pi_\theta) \quad (6)$$

$$\theta^{t+1} = \theta^t + \gamma \nabla_{\theta^t} R(s,a) \quad (7)$$

하지만 DPG에 의한 정책 업데이트는 상태의 단일 시퀀스에 대해 얻은 보상 값만을 정책 업데이트에 이용한다는 점에서 학습의 안정성이 보장되지 않는다. 그래서 본 논문은 성능향상과 안정성에 기여하기 위한 방법으로 A3C를 Deterministic Policy Gradient 방법과 결합하여 암호화폐 포트폴리오 관리 에이전트 모델에 접목시켰다.

2.2 Asynchronous Advantage actor-critic (A3C)

A3C 알고리즘은 여러 개의 에이전트가 비동기적으로 학습한다(Fig. 1 참조). 하나의 메인 쓰레드로부터 파생된 여러 개의 각 쓰레드가 독립적인 에이전트 역할을 수행한다. 메인 쓰레드는 전역 뉴럴 네트워크를 가지고 있으며 전역 뉴럴 네트워크는 정책 네트워크와 가치 네트워크로 분리된다. 각각의 에이전트도 전역 뉴럴 네트워크와 동일한 구조를 갖는 지역 뉴럴 네트워크를 소유한다. 메인 쓰레드의 전역 뉴럴 네트워크는 정책 네트워크 파라미터 벡터 θ와 가치 네트워크 파라미터 벡터 θ_v을 갖고, 에이전트의 지역 뉴럴 네트워크는 정책 네트워크 파라미터 벡터 θ'와 가치 네트워크 파라미터 벡터 θ'_v를 갖는다.

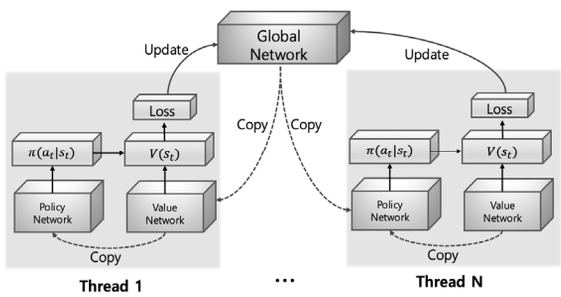

Fig. 1. A3C Model Architecture

정책 네트워크는 임의의 스텝 t에서 상태 s_t에 대한 행동 a_t 결정에 관한 정책 $\pi(a_t|s_t)$을 근사하고, 가치 네트워크는 상태 s_t에 대한 기대보상을 나타내는 가치함수 $V(s_t)$를 근사한다. 에이전트는 각 에피소드 마지막 스텝의 행동을 수행한 이후, 지역 뉴럴 네트워크에 존재하는 정책 네트워크와 가치 네트워크를 통해서 메인 쓰레드가 가진 전역 뉴럴 네트워크를 업데이트 한다. 이 후 새로운 에피소드의 학습이 수행될 때, 에이전트는 전역 뉴럴 네트워크의 파라미터를 지역 뉴럴 네트워크로 복사하여 사용한다.

한편, Advantage-Function $A(s_t, a_t)$은 현재 정책에 의해 계산된 보상과 가치함수 $V(s_t)$와의 차이를 나타내는 지표이며 정책 네트워크 업데이트에 활용되며, 다음과 같이 정의된다.

$$A(s_t, a_t) = R(s_t, a_t) - V(s_t) \quad (8)$$

Equation (8)에서는 $A(s_t, a_t)$의 분산을 줄이기 위해 보상 감쇠율 γ를 활용하여 감쇠된 누적 보상에서 실제 가치 $V(s_t)$를 빼주는 방법을 활용함을 알 수 있다.

지역 뉴럴 네트워크 내 정책 네트워크와 가치 네트워크의 Loss-Function은 다음과 같고,

$$\nabla_{\theta'} \log \pi(a_t|s_t)(R(s_t, a_t) - V(s_t)) \quad (9)$$

$$d(R(s_t, a_t) - V(s_t))^2 / d\theta'_v \quad (10)$$

정책 네트워크와 가치 네트워크의 파라미터 벡터의 업데이트 수식은 다음과 같다[9].

$$d\theta = d\theta + \nabla_{\theta'} \log\pi(a_t|s_t)(R(s_t, a_t) - V(s_t)) \quad (11)$$

$$d\theta_v = d\theta_v + d(R(s_t, a_t) - V(s_t))^2 / d\theta'_v \quad (12)$$

Cross-Entropy의 MLE (Maximum Likelihood Estimation) 이론에 따르면 미래의 결과를 알 수 없는 상황에서는 정책 네트워크의 보상을 이용해서 불확실도를 낮추는 정책을 찾는다[16, 17]. 타깃 One-Hot 벡터 d_n을 $[d_{n1}, d_{n2}, \cdots, d_{nK}]^T$로 정의하고 y_k를 k번째 행동의 확률로 정의할 때 MLE $L(\theta)$의 수식은 다음과 같다 [17].

$$L(\theta) = \prod_{n=1}^{N} p(d_n|x_n;\theta) = \prod_{n=1}^{N}\prod_{k=1}^{K} (y_k|x_n;\theta)^{d_{nk}} \quad (13)$$

위 Equation (13)를 활용한 Cross-Entropy $E(\theta)$는 다음과 같이 정의되며, 최종적으로 $E(\theta)$는 Equation (11)에서의 Loss-Function을 구성하는 $\log\pi(a_t|s_t)$ 대신 사용된다.

$$E(\theta) = -\log L(\theta) \quad (14)$$

2.3 포트폴리오 투자 관리

암호화폐 포트폴리오 관리 문제는 소유한 자산을 투자 가능한 암호화폐에 지속적인 분산투자를 하여 초기 투자 자산에 대비해 더 많은 자산을 얻는 것을 목표로 한다. 이와 관련된 선행 연구로서는 SCRP (Successive Constant Rebalanced Portfolios) [18], ONS (Online Newton Step algorithm) [19] 등이 있다. 이러한 연구들에서는 Price-Relative Vector를 정의하여 사용한다. $v_{i,t}$를 암호화폐 i에 대한 스텝 t에서의 종가로 정의하고 v_t를 전체 m개의 암호화폐에 대한 스텝 t에서의 종가 벡터라고 정의할 때, 스텝 t에서의 Price-Relative Vector y_t는 시간 t에서의 v_t에 대해 피이전 시간의 v_{t-1}로 나누어 준 것으로 다음 수식과 같다[11].

$$y_t = v_t \oslash v_{t-1} = [1.0, \frac{v_{1,t}}{v_{1,t-1}}, \frac{v_{2,t}}{v_{2,t-1}}, \cdots, \frac{v_{m,t}}{v_{m,t-1}}]^T \quad (15)$$

위 Equation (15)에서 y_1은 $[1.0, 1.0, 1.0, \cdots, 1.0]^T$로 가정하며, 본 논문에서도 Equation (15)에서 제시된 Price-Relative Vectors는 포트폴리오 관리를 위한 강화학습에서 활용한다. 하지만, 기존 연구들[18, 19]은 위와 같은 Price-Relative Vector를 이용해서 미래의 Price-Relative Vector만을 예측할 뿐, 포트폴리오 투자 관리 자체를 수행하는 방법을 제시하지 않는다. 머신러닝의 회귀기법을 이용하여 투자 전망이 좋은 금융 상품을 선별하는 기법을 제시하는 [20]을 비롯하여 딥러닝의 뉴럴 네트워크를 활용하여 금융지표 중 하나인 S&P 500을 예측 하려는 연구인 [21]도 포트폴리오 관리를 직접 수행하는 방안을 제시하지 않는다.

한편, [22-24]는 지능적인 에이전트 모델 개발을 통해 효과적으로 포트폴리오 관리를 수행한 연구 결과를 제시한다. 이러한 연구들은 에이전트의 능동적인 투자 관리를 통해 효과적인 포트폴리오 관리 방안을 제시한다. 하지만, 사람이 미리 정해놓은 패턴 조건에 일치하는 상황이 발생할 때에만 수익을 기대할 수 있다. 하지만, 강화학습에 의해 학습된 에이전트는 사람이 정할 수 있는 패턴 이외에 상황 속에 잠재된 또 다른 패턴을 새로 찾아낼 수 있다.

최근에 강화학습 에이전트가 암호화폐에 대하여 포트폴리오를 지능적으로 관리하는 연구가 [11]에서 제시되었으며, 본 논문이 제안하는 알고리즘의 기초 모델인 DPG 기법을 통하여 암호화폐를 분산 투자하여 수익을 향상시킬 수 있음을 보였다.

본 논문은 최근 대표적인 강화학습 알고리즘으로 주목받고 있는 A3C 알고리즘을 [11]에서 제시된 DPG에 접목하여 암호화폐 분산 투자 수익률을 더욱 높이는 방안을 제시한다.

3. 데이터 처리

본 논문에서 기준 통화는 원화로 정하며, 국내 빗썸 거래소에서 원화 기반의 암호화폐 가격 데이터를 취득 및 관리한다. 빗썸 API [25]를 활용해 거래 가능한 모든 암호화폐의 시가(Opening Price), 종가(Closing Price), 고가(High Price), 저가(Low Price), 타임스탬프(Timestamp) 데이터를 단위 시간 T 마다 수집한다. 이 때 의도치 않은 문제를 포함한 비정상 데이터가 수집될 수 있다. 본 장에서는 이러한 비정상 데이터를 정상 데이터로 전처리(Preprocessing)하는 방식과 그러한 데이터를 강화 학습에 사용될 데이터셋으로 가공하는 방법을 설명한다.

3.1 비정상 데이터 전처리

정상 데이터는 강화학습 데이터셋 구성을 위하여 곧바로 활용할 수 있는 데이터이다. 반면 비정상 데이터에는 부적합 데이터, 손실 데이터, 제로 데이터가 존재한다. 부적합 데이터란 데이터가 단위 시간 T 간격으로 수집되지 않은 데이터를 말한다. 손실 데이터는 단위 시간 T 간격으로 수집되었으나 실제 내용이 비어 있는 경우이다(Table 1의 5, 6번째 행). 마지막으로 제로 데이터는 단위 시간 T 간격으로 수집되었으나 가격 정보가 '0'으로 기록되어있는 경우이다(Table 1의 2, 3번째 행의 Closing Price).

이러한 비정상 데이터의 출현 빈도는 정상 데이터 출현 빈도에 비해 매우 작지만, 비정상 데이터에 대한 전처리가 올바르게 수행되지 않으면 이후 강화학습 성능에 영향을 미친다. Table 1은 단위 시간 T가 10분으로 수집된 암호화폐 BTC (Bitcoin)에서 발생한 비정상 데이터의 예시이다. 아래 1)~3)에서는 Table 1을 예로 들어 비정상 데이터에 대한 처리 방법에 대해 설명한다.

Table 1. Example of Abnormal Data in the Collected BTC Data

	Timestamp	Date & Time (String)	Opening Price	Closing Price	High Price	Low Price
1	1541951400000	2018.11.12 00:50	7,200,000	7,200,000	7,200,000	7,198,000
2	**1541951940000**	**2018.11.12 00:59**	7,198,000	**0**	7,210,000	7,198,000
3	1541952600000	2018.11.12 01:10	7,203,000	**0**	7,204,000	7,200,000
4	1541953200000	2018.11.12 01:20	7,204,000	7,206,000	7,206,000	7,200,000
5	1541953800000	2018.11.12 01:30				
6	1541954400000	2018.11.12 01:40				
7	...					
8	1541968200000	2018.11.12 05:30	7,230,000	7,229,000	7,234,000	7,229,000
9	1541968800000	2018.11.12 05:40	7,231,000	7,232,000	7,234,000	7,230,000
10	**1541969940000**	**2018.11.12 05:59**	7,236,000	7,234,000	7,236,000	7,233,000

1) 부적합 데이터 처리

매 10분 정각마다 암호화폐 데이터를 수집할 때, API를 통해 가져오는 데이터의 타임스탬프가 10분 정각의 데이터가 아닐 수가 있다. 이러한 경우, 단순하게 단위 시간 10분 기준으로 잘못된 타임스탬프 값을 수정한다. Table 1의 2, 10번째 행이 그에 대한 예시이며, '2018.11.12 00:59'는 '2018.11.12. 01:00'으로, '2018.11.12 05:59'는 '2018.11.12. 06:00'으로 변경한다 (Table 1의 Timestamp도 이에 따라 함께 변경한다).

2) 손실 및 제로 데이터 처리

수집된 임의의 데이터 내에 연속된 k개의 가격 정보가 없거나 0인 경우에, 바로 이 전 가장 가까운 올바른 데이터 정보를 X_a라고 하고 바로 이 후 가장 가까운 올바른 데이터의 정보를 X_{a+k+1}라고 할 때, 유실된 가격 정보 X_{a+i} $(1 \leq i \leq k)$는 다음 수식과 같이 보정된다.

$$X_{a+i} = X_a + i \times \frac{X_{a+k+1} - X_a}{k+1} \tag{16}$$

손실 데이터와 제로 데이터의 전처리 방법은 같고, 제로 데이터의 수정방법을 예로 들면, Table 1의 2, 3번째 행의 종가는 연속해서 제로 데이터가 수집되었다. 이 경우 Equation (16)을 활용하여 데이터를 수정한다. X_a는 7,200,000이고, k는 2이고 X_{a+3}은 7,206,000이다. X_{a+1}인 2번째 행의 종가는 0에서 7,202,000으로, X_{a+2}인 3번째 행의 종가는 0에서 7,204,000으로 치환된다.

3.2 데이터셋 정의

비정상 데이터가 앞서 설명된 전처리 방법으로 처리된 이후에는 암호화폐의 시가를 제외한 종가, 고가, 저가 데이터를 활용하여 데이터셋이 구성된다. 암호화폐 데이터셋은 시계열 데이터이며, 단위 시간 T를 기준으로 형성되는 임의의 시점 t의 종가가 대부분의 경우 시점 $t+1$의 시가와 동일하기 때문에 데이터셋에서 시가를 제외하였다. 총 m개의 암호화폐에 대하여, T를 기준으로 과거 n개의 종가, 고가, 저가 데이터를 활용하여 데이터셋을 구성한다. 이 때, n은 윈도우 크기(Window Size)라고 일컫는다. 따라서 임의의 시점 t의 데이터셋 X_t의 차원 구성(shape)은 $(3, m, n)$이다. 이후 5장에서 알 수 있듯이 본 논문에서 단위 시간 T는 기본적으로 10분으로 정한다.

한편, $v_{i,t}$, $v_{i,t}^{high}$, $v_{i,t}^{low}$는 각각 i번째 암호화폐의 임의의 시점 t에서의 종가, 고가, 저가로 정의한다. 기준 통화(즉, $i=0$)인 원화는 가격의 변동이 없기 때문에 다음 수식이 성립한다.

$$v_{0,t} = v_{0,t}^{high} = v_{0,t}^{low} = 1 \tag{17}$$

또한, V_t, V_t^{high}, V_t^{low} 벡터는 m개의 암호화폐에 대한 임의의 시점 t를 기준으로 정규화된 종가, 고가, 저가 벡터이다. 각각의 벡터에 대하여, 윈도우 크기 n에 대한 마지막 종가 대비 벡터 요소별 나누기 연산(⊘)을 통해 벡터 내에 존재하는 기존 가격 데이터를 다음과 같이 정규화한다.

$$V_{m,t} = [v_{m,t-n+1} \oslash v_{m,t} | v_{m,t-n+1} \oslash v_{m,t} | \cdots | v_{m,t-1} \oslash v_{m,t} | 1],\ t \geq n \tag{18}$$

$$V_{m,t}^{high} = [v_{m,t-n+1}^{high} \oslash v_{m,t}^{high} | v_{m,t-n+1}^{high} \oslash v_{m,t}^{high} | \cdots | v_{m,t-1}^{high} \oslash v_{m,t}^{high} | 1],\ t \geq n \tag{19}$$

$$V_{m,t}^{low} = [v_{m,t-n+1}^{low} \oslash v_{m,t}^{low} | v_{m,t-n+1}^{low} \oslash v_{m,t}^{low} | \cdots | v_{m,t-1}^{low} \oslash v_{m,t}^{low} | 1],\ t \geq n \tag{20}$$

Fig. 2는 임의의 시점 t에서의 X_t를 도식화한 것이다.

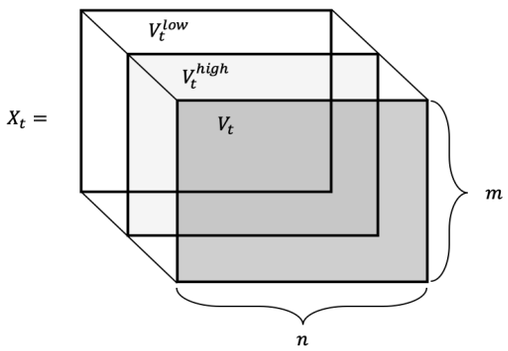

Fig. 2. Structure of Dataset X_t for Window Size n

4. 강화학습 모델

4.1 보상 정의

본 논문에서 제안하는 포트폴리오 관리 대상 암호 화폐는 사전에 m개를 미리 정한다고 가정한다. 본 모델에서 가정하는 시간은 시점 0부터 시작하며 암호화폐 자산 분배 행동은 단위 시간 T를 기준으로 형성되는 매 시점 t ($t \geq 1$)마다 강화학습의 스텝 t가 수행된다. 임의의 스텝 t에서의 가중치 벡터(Weight Vector) w_t는 각 암호화폐별 자산 분배율 벡터로 정의되며, 이 벡터를 구성하는 요소 $w_{i,t}$는 스텝 t에서의 암호화폐 i의 자산 분배율이다($0 \leq w_{i,t} \leq 1$). $w_{0,t}$는 스텝 t에서 전체 보유 자산 대비 기준 통화(원화)의 분배율을 의미한다. 포트폴리오 관리 시작 시 보유 자산은 모두 기준 통화로 가지고 있으며($w_{0,0} = 1$), 아래 수식과 같이 임의의 스텝 t에 대해 기준 통화를 포함하여 각 암호화폐 자산 분배율의 총합은 항상 1이다.

$$\sum_{i=0}^{m} w_{i,t} = 1 \tag{21}$$

에이전트는 자산 분배율에 따라 암호화폐를 매매하며, 스텝 $t-1$에서 m개의 암호화폐에 대한 종가 벡터 v_{t-1}가 스텝 t에서 v_t로 변동됨에 따라서 각 암호화폐의 잔고가 변경된다. 각 암호화폐의 보유 잔고는 스텝 t에서 암호화폐 m개에 대한 보유 수량을 의미하며 b_t로 표현된다.

스텝 t에서 자산 분배 전 변동 포트폴리오 자산 가치 p'_{t-1}는 단위 시간동안 변동된 암호화폐 가격에 따른 변화가 계산된 보유 자산 총액이다. 수식으로는 다음과 같다(연산자 $\bullet$는 두 벡터간의 내적을 의미한다).

$$p'_{t-1} = v_t \bullet b_{t-1} \tag{22}$$

한편 자산 분배 시에 따른 수수료가 지불되기 때문에, 자산 분배 전 변동 포트폴리오 자산 가치에서 분배되는 암호화폐의 분배량에 따라 수수료가 지불된다. 거래되는 암호화폐의 양에 따라 수수료비율 c가 적용되어 수수료가 계산된다. 스텝 t에서 수수료가 지불된 이후 포트폴리오 자산 가치 p_t와 보유 잔고 b_t는 다음과 같이 정의된다(연산자 $\odot$는 두 벡터의 동일 위치 원소들의 곱셈을 의미한다).

$$p_t = p'_{t-1} - c\sum_{i=1}^{m} |v_t \odot b_{t-1} - p'_{t-1} w_t| \tag{23}$$

$$b_t = p_t w_t \oslash v_t \tag{24}$$

한편, 스텝 t에서 로그 손익률 r_t는 다음과 같이 표현 가능하다.

$$r_t = \ln(p_t / p_{t-1}) \tag{25}$$

강화학습 에이전트가 스텝 t에서 자산을 분배시키는 행동을 한 후에 받는 보상이 바로 Equation (25)에 기술된 r_t가 된다.

Fig. 3은 임의로 선정한 암호화폐 3종에 대해서 시점 0 이후 3번의 스텝에 걸쳐서 자산을 분배시키는 에이전트의 보상 r_t의 산출 과정을 상세한 예시로서 보여준다. 이 예에서 수수료 비율 c는 거래와 판매 시 동일하게 0.0015로 가정한다. 시점 0에서 보유한 포트폴리오 자산 가치는 10,000이다. 이 후 스텝 $t=1$에서 포트폴리오 가치는 9,985이고, 로그 손익률 r_t는 -0.0015이다. 이후 스텝 $t=2$에서 포트폴리오 자산 가치는 10,180으로 증대되었고, 로그 손익률 r_t는 0.0193로 긍정적인 보상이 나왔다. 마지막 스텝 $t=3$에서는 포트폴리오 자산 가치가 10,089로 소폭 감소되었고, 로그 손익률 r_t은 -0.0090으로 계산되었다.

4.2 Mini-Batch Sampling

강화학습 에이전트의 궁극적인 목표는 마지막 스텝 t_f에서 최종 포트폴리오 자산 가치를 최대화 시키는 것이다. 시계열 데이터의 특성상 입력 데이터의 수는 곧 기간을 의미한다. 즉 학습에 입력되는 데이터 수에 따라서 모델의 성능이 달라질 수 있다. 그래서 본 연구에서는 확률적 경사하강법에 의한 학습 효율을 저해하지 않는 수준에서 전체 학습 데이터를 작게 나누어 입력 데이터로 활용하는 미니배치 샘플링(Mini-Batch Sampling) 방식을 사용했다[26]. 샘플링 잡음(Sampling-Noise)과 샘플링 편향(Sampling-Bias) 문제를 피하기 위해서, 미니배치 샘플링 방식에 다음 수식과 같은 기하분포 확률(Geometrically Distributed Probability) $P_\beta(b)$를 이용하여 임의의 스텝 t에서의 샘플 데이터의 시작 스텝 b를 정하였다.

$$P_\beta(b) = \beta(1-\beta)^{t-b-n} \tag{26}$$

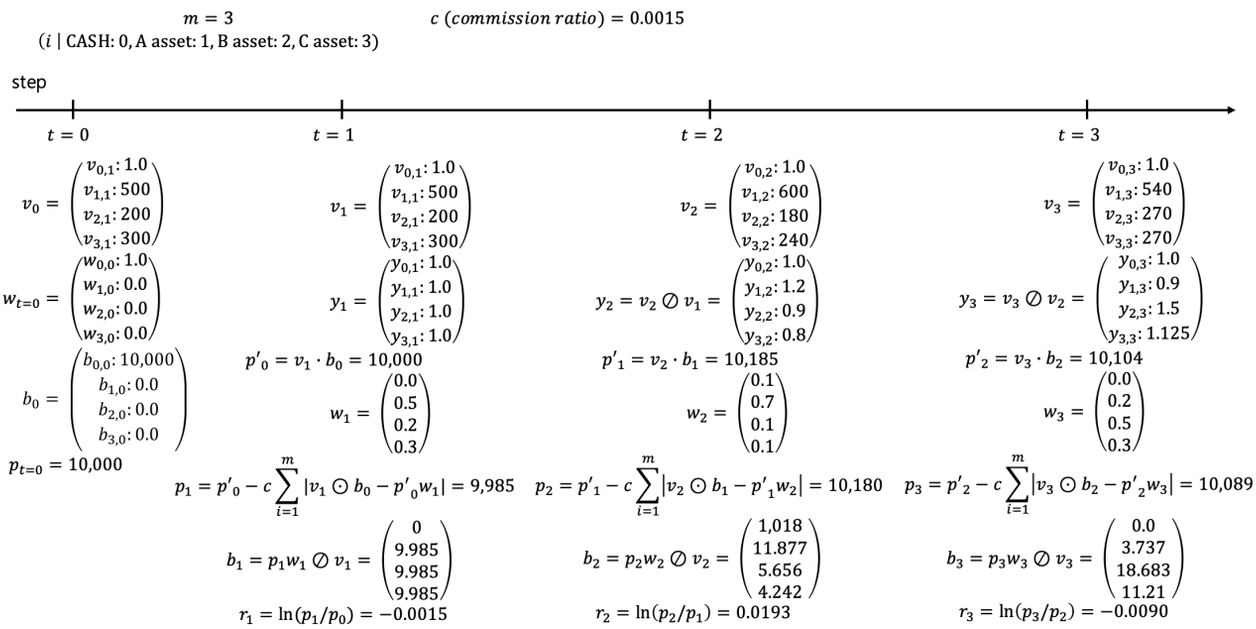

Fig. 3. Portfolio Management Process Example

즉, 미니배치 샘플링을 할 때에 $P_\beta(b)$에 근사하여 $b \leq t-n$를 만족하면서 무작위로 선택된 b 및 미리 정의한 윈도우 크기 n에 대하여 $[b, b+n]$ 스텝 범위의 데이터를 미니 배치 데이터로 사용한다. 한편, Equation (26)에 활용되는 계수 $\beta \in (0,1)$는 기하 분포의 모양을 결정하며, 이에 따라 시장 내 어떤 시기의 데이터를 더 중요시하는지 조정할 수 있다.

에이전트는 미니배치 샘플링 방식을 학습과 더불어 온라인 학습에서도 사용한다. 새로운 데이터가 수집될 때마다 에이전트는 그 것들을 유지시킴으로써 추후에 학습데이터로 사용하는데, 이를 통해 평가가 이루어지고 있는 모델일지라도 에이전트가 지속적으로 학습시킬 수 있다.

4.3 Deterministic Policy Gradient based A3C Model

본 절에서는 제안하는 모델 Deterministic Policy Gradient based A3C (A3C-DPG)에 대한 설명과 기존 DPG 기법과의 차이점에 대해 설명한다.

대부분의 강화학습은 뉴럴 네트워크 모델을 거쳐 나온 결과 값 중에서 가장 값이 큰 행동을 선택한다. 하지만 본 논문의 에이전트는 매 스텝 마다 각 원소의 총 합이 1인 확률 벡터 $[p[a_1|s,\theta], p[a_2|s,\theta], \cdots, p[a_n|s,\theta]]$ 자체를 에이전트 행동, 즉 4.1절에서 정의한 임의의 스텝에서의 w_t로 활용한다. 따라서 모델을 거쳐 나온 행동을 Cross-Entropy 수식 Equation (13)의 인자 d_n과 같은 One-hot 벡터 형태로 나타낼 수 없다. 이 문제를 해결하기 위해서 Equation (13)의 $p(d_n|x_n;\theta)$ 대신에 에이전트 행동에 해당하는 $[p[a_1|s,\theta], p[a_2|s,\theta], \cdots, p[a_n|s,\theta]]$의 원소 각각의 제곱을 모두 더한 값을 대입하였다. 정책 네트워크 모델을 통하여 얻은 행동 벡터의 k번 째 원소 y_k를 $p[a_k|s,\theta]$로 정의할 때, 변형된 MLE $L'(\theta)$ 및 Cross-Entropy $E'(\theta)$을 다시 정리하면 다음과 같다.

$$\begin{aligned} L'(\theta) &= \prod_{n=1}^{N}\sum_{k=1}^{K} p(a_k|s_n;\theta)^2 \\ &= \prod_{n=1}^{N}\sum_{k=1}^{K} (y_k|s_n;\theta)^2 \end{aligned} \tag{27}$$

Table 2. Data ranges for back-test

Market Conditions	ID	T	Learning data period	Back-Test data period
General Market	Back-Test #1-1	10	2017-09-29 00:10:00 ~ 2018-09-21 00:00:00	2018-09-21 00:10:00 ~ 2018-10-06 00:00:00
	Back-Test #1-2	30	2017-09-29 00:30:00 ~ 2018-09-21 00:00:00	2018-09-21 00:30:00 ~ 2018-10-06 00:00:00
Bull Market	Back-Test #2-1	10	2017-09-29 00:10:00 ~ 2018-02-05 00:00:00	2018-02-05 00:10:00 ~ 2018-02-21 00:00:00
	Back-Test #2-2	30	2017-09-29 00:30:00 ~ 2018-02-05 00:00:00	2018-02-05 00:30:00 ~ 2018-02-21 00:00:00
Bear Market	Back-Test #3-1	10	2017-09-29 00:10:00 ~ 2018-03-01 00:00:00	2018-03-01 00:10:00 ~ 2018-03-16 00:00:00
	Back-Test #3-2	30	2017-09-29 00:30:00 ~ 2018-03-01 00:00:00	2018-03-01 00:30:00 ~ 2018-03-16 00:00:00

$$E'(\theta) = -\log L'(\theta) \tag{28}$$

에이전트에게 주어진 보상 체계는 Full-Exploitation 방식으로 정하여 항상 정책 네트워크에서 제시하는 포트폴리오 분배 가중치에 따라 자산을 분배한다. 매 스텝마다의 상태와 행동을 모두 고려하는 보상 함수를 정책 매개변수 업데이트에 사용하며, 스텝 f가 종료된 시점의 에피소드의 최종 보상 함수 R_f는 다음과 같다.

$$R_f = \frac{1}{f} ln \frac{p_f}{p_0} = \frac{1}{f} \sum_{t=1}^{f} r_t \tag{29}$$

4.2 절에서 언급한 미니배치 샘플링 방식에 의해 샘플링된 미니 배치 데이터에 대한 Advantage-Function은 다음과 같고,

$$A(s_f, a_f) = \frac{1}{f} \sum_{t=1}^{f} r_t - r_f \tag{30}$$

정책 네트워크와 가치 네트워크의 업데이트 수식은 다음과 같다.

$$\theta \rightarrow \theta - \lambda \nabla_{\theta'} \log L'(w) A(s_f, a_f) \tag{31}$$

$$\theta_v \rightarrow \theta_v + d(A(s_f, a_f))^2 / d\theta'_v \tag{32}$$

5. 실험 평가

암호화폐 포트폴리오 관리 목적은 보유한 통화를 거래 가능한 다른 암호화폐와 교환하여 최종 포트폴리오 자산 가치를 증대시키는 것이다. 실험 평가에서는 1) 본 논문이 제시하는 모델인 A3C-DPG, 2) [11]의 연구에서 제시된 DPG, 그리고 3) 무작위로 포트폴리오 관리를 하는 Random 모델에 대한 백테스트(Back-Test)를 진행한다.

5.1 백테스트 및 강화학습 성능 평가기준

벡테스트가 종료되면 주어진 백테스트 기간의 마지막 스텝까지 누적된 포트폴리오 자산 가치 증감 비율(Portfolio Value Variation Ratio, PVVR)을 통해서 각 모델의 수익률을 비교평가 하였다. 스텝 t에서 PVVR P^t의 수식은 아래와 같으며,

$$P^t = p_t / p_{t-1} \tag{33}$$

마지막 스텝 f에서 누적된 PVVR P^f은 다음과 같다.

$$P^f = \prod_{t=1}^{f} P^t = p_f / p_0 \tag{34}$$

한편 모델이 가변적인 시장 상황에 따라 유연하게 대응할 수 있는지를 평가하기 위해서 백테스트의 기간을 '일반', '상승장', '하락장'으로 나누어 비교하였다. 그리고 단위 시간 T를 10분과 30분으로 나누어 자산의 분배 주기에 따른 모델 성능을 비교평가 하였다. 3.2절에서 설명한 임의의 데이터셋 샘플의 윈도우 크기 n은 단위 시간 기준으로 100으로 정하였다. 따라서 강화학습 모델에 입력으로 사용되는 임의의 데이터셋 샘플 1개의 시간 길이는 T가 10이면 1000분(=16시간 40분)이며, T가 30이면 3000분(=2일 2시간)이다.

백테스트는 실제 암호화폐 시장과 같은 투자 시뮬레이션 환경에서 에이전트의 포트폴리오 관리 성능을 평가하기 위해 사용된다. 본 실험에 쓰인 암호화폐의 종류는 총 8종이며 각각의 암호화폐 명칭은 BTC, ETH, XRP, BCH, ETC, DASH, XMR, ZEC이다. 이들은 비교적 일찍 빗썸 거래소에 상장되고 거래량도 많은 암호화폐들이다. Table 2는 학습에 사용된 학습 데이터 기간과 백테스트를 위한 데이터 기간을 제시한다. 학습이 진행되는 각 에피소드 마다 미니배치 샘플링 되는 전체 데이터 기간은 2일이며, 2일이 지나면 PVVR이 산출된다. 학습이 종료되면 에이전트는 학습 데이터 기간 직후 바로 이어지는 15일 간의 백테스트를 진행한다. 백테스트가 시작되면 에이전트는 학습과 동일한 방식으로, T 만큼의 시간이 지나는 매 스텝 t마다 포트폴리오 재분배를 하게 된다. 백테스트가 종료된 후 산출된 최종 PVVR P^f를 통해 모델의 성능을 측정한다.

게다가, PVVR 이외에 백테스트 종료 시에 계산되는 추적오차(Tracking Error, TE)와 정보비율(Information Ratio, IR) 지표를 통해서 기존 DPG 방식 대비 A3C-DPG의 성능 향상 정도를 추가적으로 비교분석하였다 [27, 28]. 추적오차는 백테스트 기간 내 각 스텝 t마다의 포트폴리오 가치 증감 비율인 $P^t_{A3C-DPG}$과 P^t_{DPG}를 비교함으로써 DPG와 A3C-DPG 모델 간의의 성능격차를 측정한다. 다음은 실험에 사용된 추적오차 계산 수식이다.

$$TE = \sqrt{\frac{\sum_{t=1}^{f} (R^t_{A3C-DPG} - R^t_{DPG})^2}{N-1}} \tag{35}$$

추적오차가 작다면 두 모델이 비슷한 행동을 한다는 것을 암시하고 반대로 커지게 된다면 그 만큼 두 모델이 다른 행동을 하며 그에 따른 차이가 발생한다는 의미이다.

한편, A3C-DPG 모델이 DPG 모델보다 더 좋은 성과를 내기 때문에 추적오차의 차이가 생기는 것인지 아니면 그 반대의 경우 때문에 차이가 생기는 것인지는 정보비율 지표로써 알아볼 수 있다. 정보비율은 DPG 모델 대비 A3C-DPG 모델의 최종 초과 투자 수익 $P^f_{A3C-DPG} - P^f_{DPG}$을 추적오차로 나누어 준 것이며, 백테스트에서 제시하는 모델이 기존 모델에 비해 어느 정도로 적극적인 운용을 하며 수익을 내는지 알 수 있는 지표이다. 다음은 실험에 사용된 정보비율 계산 수식이다.

$$IR = \frac{P^f_{A3C-DPG} - P^f_{DPG}}{TE} \tag{36}$$

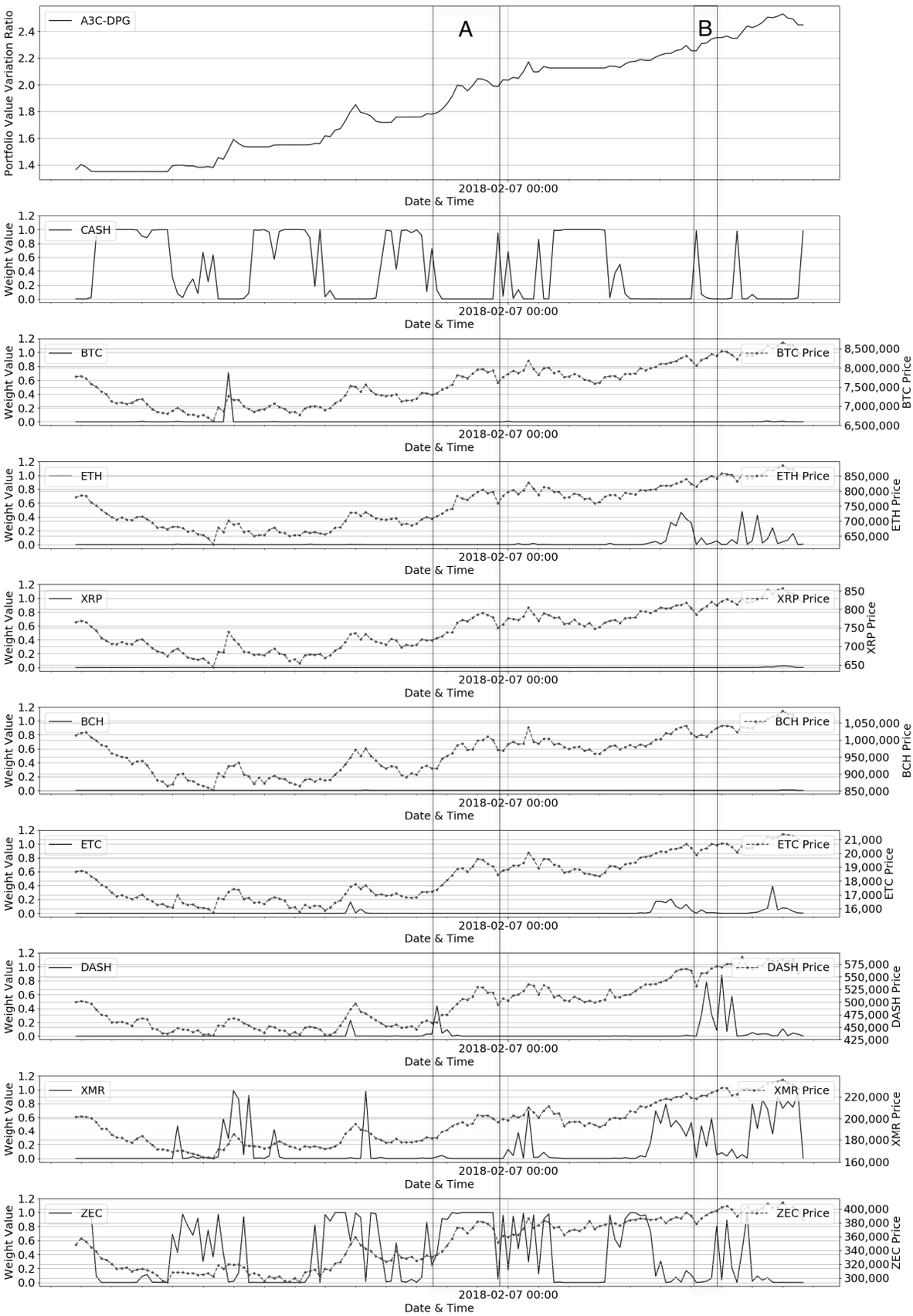

Fig. 4. Weight Distribution Ratio of Back-Test #3-2

5.2 모델 성능 비교 결과

Table 3. The Final PVVR Comparison between Models A3C-DPG, DPG, and Random

	Final Portfolio Value Variation Ratio		
	A3C-DPG	DPG	Random
Back-Test #1-1	2.4271	1.0840	1.1012
Back-Test #1-2	1.6007	1.4968	1.0896
Back-Test #2-1	17.3247	4.5705	1.6175
Back-Test #2-2	4.4227	2.7413	1.3524
Back-Test #3-1	1.8283	1.2304	0.7155
Back-Test #3-2	1.4632	1.3918	0.7201

제시된 Table 3에서 보이듯이, 본 논문에서 제안하는 A3C-DPG 모델이 Random 모델뿐만 아니라 기존 모델인 DPG 보다 전체적으로 더 높은 최종 PVVR을 산출함을 알 수 있다. 또한, 제안하는 A3C-DPG 모델인 경우 T가 30분인 경우 보다는 10분인 경우에 최종 PVVR이 높다. 즉, A3C-DPG 모델은 짧은 주기의 투자 운용을 수행하여 시장 상황에 더 민첩하게 대응할 수 있음을 암시한다. 또한, 가격 변동이 상승세를 타며 투자하기 좋은 시장상황에서는 T가 10분일 때 A3C-DPG 모델의 최종 PVVR은 17.3247로서 최초 자산 대비 17배 이상의 많은 수익을 거두었음을 알 수 있다. 또한, 가격 변동이 하락세를 타며 투자하기 좋지 못한 시장상황에서도 T가 10분일 때 A3C-DPG 모델의 최종 PVVR은 1.8283으로서 최초 자산 대비 2배 가까운 수익을 거두었음을 알 수 있다.

Table 4. The TE and IR Values of A3C-DPG Model based on DPG Model

	Tracking Error	Information Ratio
Back-Test #1-1	0.8679	1.5474
Back-Test #1-2	0.0749	1.3857
Back-Test #2-1	7.2513	1.7588
Back-Test #2-2	0.9538	1.7627
Back-Test #3-1	0.2431	2.4596
Back-Test #3-2	0.0888	0.8034

또한 백테스트에서 A3C-DPG 모델이 DPG 모델보다 더욱 활발한 투자 운용을 했음을 Table 4가 제시하는 추적오차와 정보비율 지표를 통해서 알 수 있다. Back-Test #2-1의 경우 A3C-DPG 모델이 DPG 모델보다 높은 초과 투자 수익을 얻었기 때문에 7.2513이라는 높은 추적오차가 발생했음에도 정보비율이 1.7588로 계산되었음을 알 수 있다. 반면 Back-Test #3-1은 A3C-DPG 모델이 DPG 모델보다 효율적인 투자 움직임을 보였기 때문에 0.2431의 낮은 추적오차에도 불구하고 2.4596이라는 높은 정보비율이 계산되었음을 확인할 수 있다.

5.3 자산 분배율 분석

이번 절에서는 에이전트가 백테스트에서 높은 FPV를 얻을 수 있었던 이유를 Fig. 4에서 제시하는 암호화폐 자산 분배율 분석을 통해서 알아본다. Fig. 4는 FPV가 가장 높았던 Back-Test #2-1에서 백테스트 기간 중 3일 동안의 암호화폐별 자산 분배율을 각 암호화폐의 종가와 함께 보여준다. 첫 번째 그래프는 해당 기간의 PVVR 추이를 제시하고, 두 번째 그래프는 동일 기간의 보유 자산 중 원화의 비율을 제시한다. 세 번째부터 마지막까지의 그래프는 실험에 쓰인 8개 각각의 암호화폐에 대하여 동일 기간 동안의 자산 분배 비율 추이를 보여준다.

영역 A에서 대부분의 암호화폐들은 상승세를 타고 있으며 그 중에서 ZEC는 다른 암호화폐에 비해 가격 상승곡선이 가파르다. 이 같은 상황에서 강화학습 에이전트는 전체 보유 자산을 ZEC를 구매하기 위해 사용하여 ZEC의 보유 자산 비율을 높이는 행동을 했고, ZEC의 가격 하락세가 보이는 시점부터는 ZEC에 있던 자산을 원화로 환수하는 행동을 보였다. 또한 영역 B에서도 가격 상승곡선이 가장 가파른 DASH 자산을 대부분 확보하였다가 DASH 가격 상승이 주춤해질 때 비교적 더 많은 가격 상승이 예상된 XMR 자산을 더 많이 확보하였다. 영역 A와 B에서 이와 같은 행동을 통하여 전체 PVVR을 상승시킴을 확인할 수 있다.

6. 결 론

본 논문은 전통적인 금융 포트폴리오 관리 문제를 강화학습 프레임워크를 사용하여 해결한 Deterministic Policy Gradient (DPG) 모델을 개선시킨 Deterministic Policy Gradient based A3C (A3C-DPG) 모델을 제시하였다. 빗썸 API를 통해 과거 암호화폐 가격 데이터를 수집했고 학습과 백테스트를 위한 데이터셋을 만들기 위해 전처리와 가공 과정을 거쳤다. 그리고 학습 데이터셋을 미니배치 샘플링하여 강화학습 에이전트의 학습에 활용하였다. 두 모델의 백테스트를 통해서 비교 평가 실험을 하였고, 시장금융 투자의 관점에서 추적오차와 정보비율 지표를 인용함으로써 A3C-DPG 모델이 기존 DPG 모델보다 더 적극적인 투자 성향을 보인다는 것을 입증하였다. 그리고 백테스트 결과, 강화학습 에이전트는 암호화폐 시장 상황에 구애받지 않고 적극적인 투자를 했다. 특히 상승장에서 초기 투자 자본금의 약 17.3배에 달하는 수익률을 보였고, 하락장에서도 약 1.8배의 수익률을 얻었다. 또한 에이전트의 포트폴리오 투자 비율을 분석해 제시함으로써 앞선 백테스트 결과에 대한 근거를 제시하였다.

금융 데이터는 기존 강화학습에서 다루었던 컴퓨터 게임 데이터에 비해 특수하며 각 특성 데이터 값의 해석이 상당히 중요하다. 데이터 특성 값들을 사용 목적에 맞는 기준을 세워 정규화하고, 모델의 일반화가 잘 이루진다면 강화학습을 금융 투자 도메인에서도 활용할 수 있는 가능성이 존재한다.

References

[1] Nakamoto, Satoshi, Bitcoin: A Peer-to-Peer Electronic Cash System, Cryptography Mailing list at https://metzdowd.com, 2009.

[2] "GUNBOT - Crypto Trading Bot," GUNBOT, https://www.gunbot.com, 2018.

[3] "start [ProfitTrailer Wiki]", ProfitTrailer, https://wiki.profittrailer.com/doku.php?id=start, 2018.

[4] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, Vol.10, No.3, pp.215-236, 1996.

[5] Candela, "Dataset shift in machine learning," London: MIT Press, 006.3 CAN, 2009.

[6] Y. B, Kim, "Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies," *PLoS ONE*, Vol.11, No.8, e0161197, 2016.

[7] Sean McNally, "Predicting the Price of Bitcoin Using Machine Learning," *26th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, pp. 339-343, Mar. 2018.

[8] R. Sutton and A. Barto, "Reinforcement Learning: an Introduction," MIT Press, 1998.

[9] Volodymyr Mnih, "Asynchronous Methods for Deep Reinforcement Learning," *Proceedings of the 33rd International Conference on MachineLearning*, New York, NY, USA, 2016. JMLR: W&CP volume48.

[10] Arun Nair, "Massively Parallel Methods for Deep Reinforcement Learning," at Deep Learning Workshop, International Conference on Machine Learning, Lille, France, 2015.

[11] Zhengyao Jiang, "A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem," In JMLR, 30 pages, 5 figures, 2017.

[12] Christopher JCH Watkins and Peter Dayan. "Q-Learning," *Machine Learning*, Vol.8, No.3-4, pp.279-292, 1992.

[13] Kai Arulkumaran, "A Brief Survey of Deep Reinforcement Learning," in I*EEE Signal Processing Magazine Special Issue On Deep Learning For Image Understanding*, 2017.

[14] Hado van Hasselt, "Deep Reinforcement Learning with Double Q-learning," *Proceedings of 30th AAAI Conference on Artificial Intelligence* (AAAI-16).

[15] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Ried-miller, "Deterministic Policy Gradient Algorithms," *ICML(International Conference on Machine Learning) Proceedings of the 31st*, pp.387-395, 2014.

[16] K. Chepuri, T. Homem de Mello, "Solving the vehicle routing problem with stochastic demands using the cross entropy method," Annals of Operations Research, 2004.

[17] G. Alon, D. P. Kroese, T. Raviv, and R. Y. Rubinstein, "Application of the Cross-entropy method to the buffer allocation problem in a simulation-based environment," Annals of Operations Research, 2004.

[18] Gaivoronski, "Stochastic nonstationary optimization for finding universal portfolios," in Annals of Operations Research, Vol.100, No.1, pp.165-188, 2000.

[19] Agarwal, A., "Algorithms for portfolio management basedon the newton method," in ICML, New York, NY, USA (2006)

[20] Bin Li, Peilin Zhao, Steven C. H. Hoi, and Vivekanand Gopalkrishnan. "Passive aggressive mean reversion strategy for portfolio selection," PSMR, *Machine Learning*, Vol.87, No.2, pp.221-258, 2012.

[21] Seyed Taghi Akhavan Niaki and Saeid Hoseinzade. "Forecasting S&P 500 index using artificial neural networks and design of experiments," *Journal of Industrial Engineering International*, Vol.9, No.1, p.1, 2013.

[22] Katia Sycara, K. Decker and Dajun Zeng, "Designing a Multi-Agent Portfolio Management System," *Proceedings of the AAAI Workshop on Internet Information Systems*, 1995.

[23] K. Sycara, A. Pannu, M. Willamson, Dajun Zeng, K. Decker, "Distributed intelligent agents," *IEEE Expert*, Vol.11, Issue 6, Dec. 1996.

[24] Hiroshi Takahashi, "Analyzing the Effectiveness of Investment Strategies through Agent-based Modelling: Overconfident Investment Decision Making and Passive Investment Strategies," eKNOW, *6th International Conference*, 2014.

[25] "API - Bithumb," Bithumb, https://www.bithumb.com/u1/US127, 2018.

[26] Mnih, Volodymyr, "Human-level control through deep reinforcementlearning," *Nature*, Vol.518, No.7540, pp.529-533, 2015.

[27] Mu Li, "Efficient Mini-batch Training for Stochastic Optimization," In 2014 ACM, 978-1-4503-2956-9, 2014.

[28] Edward Qian, "Active Risk And Information Ratio," *Journal of Investment Management*, Vol.2, No.3, pp.1-15, 2004.

김 주 봉

https://orcid.org/0000-0002-8234-1030

e-mail : jubong1992@gmail.com

2017년 한국기술교육대학교 컴퓨터공학부 (학사)

2017년~현 재 한국기술교육대학교 컴퓨터공학부 석사과정

관심분야 : Reinforcement Learning, Deep Learning

허 주 성

https://orcid.org/0000-0002-2486-9515
e-mail : chill207@koreatech.ac.kr
2016년 한국기술교육대학교 컴퓨터공학부 (학사)
2016년～현 재 한국기술교육대학교 컴퓨터공학부 석사수료
관심분야 : Machine Learning, Social Network Analysis

임 현 교

https://orcid.org/0000-0002-8807-1158
e-mail : glenn89@koreatech.ac.kr
2017년 한국기술교육대학교 컴퓨터공학부 (석사)
2017년～현 재 한국기술교육대학교 창의융합공학협동과정 ICT 융합 박사과정
관심분야 : Reinforcement Learning, Deep Learning, SDN, Network Mobility Management, Future Internet

권 도 형

https://orcid.org/0000-0002-5951-2081
e-mail : dohk@koreatech.ac.kr
2017년 한국기술교육대학교 컴퓨터공학부 (학사)
2017년～현 재 한국기술교육대학교 창의융합공학협동과정 ICT 융합 석사과정
관심분야 : Machine Learning, Stock Price Prediction

한 연 희

https://orcid.org/0000-0002-5835-7972
e-mail : yhhan@koreatech.ac.kr
1998년 고려대학교 컴퓨터학과(석사)
2002년 고려대학교 컴퓨터학과(박사)
2002년 삼성종합기술원 전문연구원
2013년～2014년 미국 SUNY at Albany, Department of Computer Science 방문교수
2006년～현 재 한국기술교육대학교 컴퓨터공학부 교수
관심분야 : Internet of Things, Machine Learning, Social Network Analysis, Future Internet