



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위 논문
지도교수 한연희

비동기 심층 강화학습 기반 금융 포트폴리오 관리

Financial Portfolio Management Based on
Asynchronous Deep Reinforcement Learning

2019년 2월

한국기술교육대학교 대학원
컴퓨터공학과 컴퓨터공학전공
김 주 봉

비
동
기

심
층

강
화
학
습

기
반

금
용

포
트
폴
리
오

관
리

2
0
1
9
·
2

- 권 -
주
봉

석사학위 논문
지도교수 한연희

비동기 심층 강화학습 기반 금융 포트폴리오 관리

Financial Portfolio Management Based on
Asynchronous Deep Reinforcement Learning

2019년 2월

한국기술교육대학교 대학원
컴퓨터공학과 컴퓨터공학전공
김 주 봉

비동기 심층 강화학습 기반 금융 포트폴리오 관리

Financial Portfolio Management Based on
Asynchronous Deep Reinforcement Learning

이 논문을 컴퓨터공학 석사학위 논문으로 제출합니다

2019년 2월

한국기술교육대학교 대학원

컴퓨터공학과 컴퓨터공학전공

김 주 봉

김주봉의

컴퓨터공학 석사학위 논문을 인준함

심사위원장 조 태 훈 (인)

심사위원 오 홍 선 (인)

심사위원 한 연 희 (인)

2018년 12월

한국기술교육대학교 대학원

감 사 의 글

석사학위 논문을 마무리하고 이 글을 쓰려하니 감회가 남다릅니다. 8년간의 학사와 석사 과정을 무사히 지나올 수 있도록 가까이서 힘이 되어준 친구들, 선후배, 그리고 저의 지도교수님께 감사의 말을 드리고 싶습니다.

2011년도 입학 후 처음 만나 잊지 못할 학교생활을 같이 해주고 취업해 대견한 친구들인 김종석, 김영근, 김범영, 유민호에게 고마움을 느낍니다. 그리고 2017년도 대학원 진학 후 연구실에서 기쁨과 어려움을 같이 나누며 동고동락한 선후배인 임현교(형), 허주성(형), 권도형, 김찬명(형), 정경석 모두에게 정말 감사합니다. 연구실 후배 최호빈, 김영규, 김찬기에게도 고맙습니다. 그리고 나의 오랜 친구들 박성정, 박민정, 지강선, 유홍록에게도 정말 고맙다는 인사를 전합니다.

2년간 배움의 기회를 주시고 앞장서서 길잡이 역할을 해주신 지도교수님인 한연희 교수님, 부족했던 제게 격려와 가르침을 아끼지 않아 주신 것에 말로 표현 못 할 만큼 깊이 감사드립니다. 그리고 논문 심사와 조언을 해주신 조태훈 교수님, 오홍선 교수님께도 감사드립니다.

마지막으로 지금까지 길러주시고 분에 넘치는 사랑을 주신 어머니, 항상 지지해주시고 믿어주고 지켜봐주신 아버지, 믿음직하고 하나뿐인 형과 따뜻한 마음을 나눠준 형수님께 감사하고 사랑합니다.

끝으로, 연구를 천직으로 삼아 계속해서 노력하겠습니다. 자만하지 않고 끊임없이 배워나가겠습니다. 제 다짐의 증인이 되어주신 모든 분들께 다시 한 번 감사합니다.

2018. 12

김주봉 드림

비동기 심층 강화학습 기반 금융 포트폴리오 관리

금융투자 관리 전략 중에서 투자자의 보유 자산을 여러 금융상품에 나누어 투자하는 것을 포트폴리오 관리라고 하며, 금융 시장에서 무수히 많은 자산들이 포트폴리오를 이루면서 효율적인 투자를 만든다. 한편 딥러닝이 여러 분야에서 괄목할만한 성과를 보이면서 심층강화학습 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) 기반 모델을 사용한 효율적인 금융 포트폴리오 투자 관리 기법을 제안한다. 또한, A3C를 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 A3C 기반 강화학습 모델의 성능이 우수하다는 것을 입증하였다.

주제어 : 금융 포트폴리오 관리, 강화학습, 분산 투자, A3C, 암호화폐

목 차

국문요약	i
목 차	ii
표 목 차	iv
그림목차	v

제1장 서론

제1절 연구배경 및 개요	1
제2절 논문 구성	3

제2장 관련 연구

제1절 강화학습	4
제2절 Asynchronous Advantage Actor-Critic (A3C)	7
제3절 포트폴리오 투자 관리	10

제3장 데이터 처리

제1절 비정상 데이터 전처리	13
1. 부적합 데이터 처리	13
2. 손실 및 제로 데이터 처리	14
제2절 데이터셋 정의	15

제4장 강화학습 모델

제1절 보상 정의	17
제2절 Mini-Batch Sampling	21

제3절 Deterministic Policy Gradient based A3C Model	22
제4절 실제 시장과 투자 시뮬레이션과의 차이	24
 제5장 실험 평가	
제1절 백테스트 및 강화학습 성능 평가 기준	25
제2절 모델 성능 비교 결과	28
제3절 자산 분배율 분석	32
 제6장 결론	
참고문헌	35
Abstract	38

표 목 차

<표 3-1> 수집된 BTC 데이터에 존재하는 이상 데이터 예시	12
<표 5-1> 백테스트를 위한 데이터의 기간	25
<표 5-2> A3C-DPG, DPG, Random 모델 간 최종 PVVR 비교	28
<표 5-3> A3C-DPG와 DPG 모델 간 추적오차와 정보비율	28

그 립 목 차

[그림 2-1] A3C 모델의 구성 요소	7
[그림 3-1] 윈도우 크기 n 에 대한 데이터셋 X_t 의 구조	15
[그림 4-1] 포트폴리오 관리 과정 예시	20
[그림 5-1] Back-Test #1-1: 일반적인 시장 상황에서의 포트폴리오 관리, $T=10$ 분, (2018-09-21 00/10/00 ~ 2018-10-06 00/00/00)	30
[그림 5-2] Back-Test #1-2: 일반적인 시장 상황에서의 포트폴리오 관리, $T=30$ 분, (2018-09-21 00/30/00 ~ 2018-10-06 00/00/00)	30
[그림 5-3] Back-Test #2-1: 상승장에서의 포트폴리오 관리, $T=10$ 분, (2018-02-05 00/10/00 ~ 2018-02-21 00/00/00)	30
[그림 5-4] Back-Test #2-2: 상승장에서의 포트폴리오 관리, $T=30$ 분, (2018-02-05 00/30/00 ~ 2018-02-21 00/00/00)	30
[그림 5-5] Back-Test #3-1: 하락장에서의 포트폴리오 관리, $T=10$ 분, (2018-03-01 00/10/00 ~ 2018-03-16 00/00/00)	31
[그림 5-6] Back-Test #3-2: 하락장에서의 포트폴리오 관리, $T=30$ 분, (2018-03-01 00/10/00 ~ 2018-03-16 00/00/00)	31
[그림 5-7] 백테스트에서의 자산 분배 비율	33

제1장 서론

제1절. 연구목적과 연구방법론

금융투자 전략 중 하나인 포트폴리오 투자 전략은 기존의 투자 방식의 위험을 감소시키고자 금융자산을 분산 투자하는 것을 일컫는다. 이 전략은 시세 변화의 예측이 어려운 금융 시장에서도 안정적으로 투자를 할 수 있도록 만들어 준다. 최근에, 그 해결책으로 주식투자 전략을 알고리즘화한 봇을 사용하고자 하는 노력이 많이 이루어지고 있다. 그 예들로 2016년 12월에 공개된 Gunbot [1]과 2017년 9월에 공개된 Profit Trailer Bot [2]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 괄목할만한 성과를 보이면서 머신러닝으로 금융 거래를 시도하는 연구가 많이 진행되어 왔다. 하지만 대부분의 연구는 정량적인 주식 가격의 흐름을 예측하는 것들이 대부분이며 알고리즘 내부에서도 주식이론을 많이 도입하고 있다 [3, 4]. 또한 제안하는 알고리즘의 성능은 예측 정확도에만 의존하며, 그러한 예측 정확도에 입각하여 거래 시그널을 생성하는 방향으로 연구가 이루어지고 있다 [5, 6].

최근 강화학습은 신경망과 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)망과 가치(Value)망을 분리하여 사용하는 A2C (Advantage

Actor-Critic), A3C (Asynchronous Advantage Actor-Critic)와 같은 강화 학습 알고리즘들이 최근 발표되고 그 성능과 효율이 주로 컴퓨터 게임 분야에서 입증된 바 있다 [7, 8, 9].

그리고 최근 강화학습을 포트폴리오 관리 전략에 활용하는 시도가 학계에서 발표되고 있다. 강화학습 에이전트가 대신 전략을 고안 하도록 하여 투자의 경험이 없거나 투자지식이 적은 사람도 높은 투자 수익을 얻을 수 있도록 함이 본 연구들의 목표이다. 이중 가장 대표적인 연구[10]은 Deterministic Policy Gradient 알고리즘을 사용한 것이다. 이 논문에서는 Ensemble of Identical Independent Evaluators (EIIE) 신경망 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 금융자산 포트폴리오 투자 전략에 최신 강화학습 알고리즘인 A3C를 접목한 연구 수행 결과를 제시한다. 본 논문에서 포트폴리오 투자 전략에 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 [6, 7]. 두 번째는 에이전트의 투자 종목 선택 및 배분을 결정에 직접 관련된 정책망과 투자 수익 성능 평가에 관련된 가치망을 분리시켜서, DQN과 다르게 그러한 분리된 정책의 가치 판단을 통하여 최종 목적인 정책에 대한 학습을 보다 효율적으로 수행할 수 있기 때문이다 [8].

한편, 금융자산 포트폴리오 관리에 A3C를 접목시키는 과정에서 기존의 Cross-Entropy 함수를 그대로 적용할 수 없기 때문에 기존의 Cross-Entropy를 변형하여 정책망 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.

제2절. 논문 구성

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. 6장에서는 전체적인 요약과 함께 결론을 제시한다.

제2장 관련 연구

제1절. 강화학습

강화학습의 목표는 에이전트가 환경과 상호작용하며 기대 누적 보상 $E[R]$ 을 최대한 높이는 것이다. 에이전트는 임의의 스텝 t 에서 환경의 상태(State) s_t 를 관찰하고, 파라미터 집합 θ 를 지닌 모델의 정책 π_θ 를 통해 도출한 행동(Action) a_t 을 수행하고, 그 행동에 대한 보상(Reward) r_t 를 받는다. 한편, 에이전트는 임의의 상태 t 에서 시작하여 T 스텝에서 에피소드가 끝난다고 가정할 때, 보상 감쇠율(Discount Factor) γ ($\gamma \in (0,1]$)를 고려한 다음 수식 (1)과 같은 상태 t 에서의 누적 보상을 얻는다 [8, 9].

$$R_t = \sum_{t'=t}^T \gamma^{t'-t} r_{t'} \quad (1)$$

Action-Value 함수 $Q^{\pi_\theta}(s_t, a_t)$ 는 현재 상태 s_t 를 관찰한 뒤, 현재 정책 π_θ 에 따른 행동 a_t 에 대한 누적보상 R_t 의 기댓값이다. 최적의 Action-Value 함수에 대한 수식은 다음과 같다.

$$Q^*(s_t, a_t) = \operatorname{argmax}_{\pi_\theta} Q^{\pi_\theta}(s_t, a_t) \quad (2)$$

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

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

Equation (3)은 강화학습의 목적인 누적보상 R_t 을 최대한 높이는 데 사용

된다 [11, 12, 13].

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 t 에서의 정책 π^t 을 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[14].

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

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.

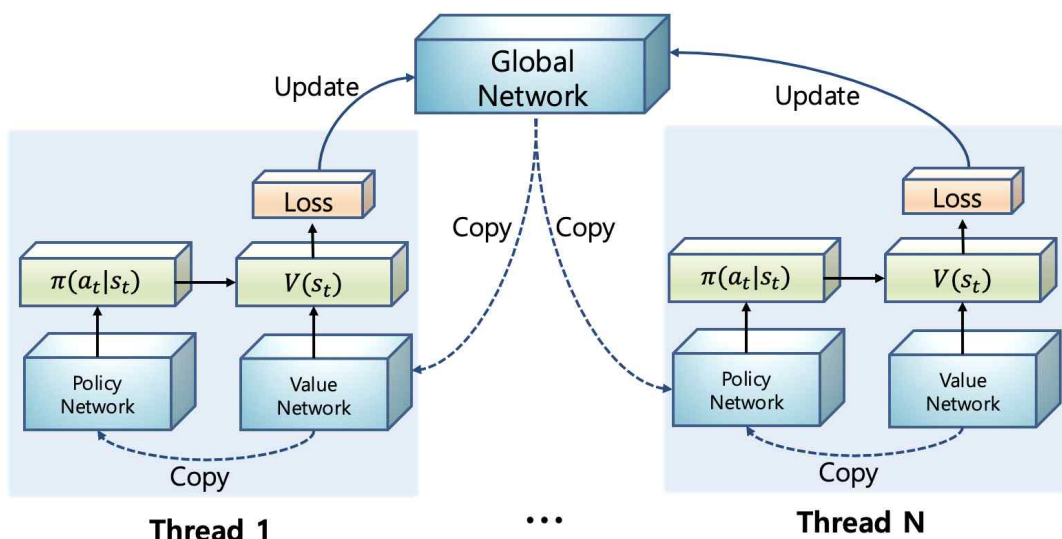
그래서 스텝이 진행 중일 때 정책 파라미터들을 업데이트하지 않고 정책에 기반 하여 모든 상태 값들을 고려하면서 업데이트하는 방법이 DPG (Deterministic Policy Gradient) 방법이다. 이 방법에서는 정책 파라미터 θ 에 대하여 행동 $a_t = \pi_{\theta}(s_t)$ 을 정의하며, 정책에 의한 행동의 보상 값을 정책의 기울기 방향으로 업데이트 시킨다. 수식으로는 다음과 같다 [10, 14].

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

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

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 금융자산 포트폴리오 관리 에이전트 모델에 접목시켰다.

제2절. Asynchronous Advantage Actor-Critic (A3C)



[그림 2-1] A3C 모델의 구성 요소

A3C 알고리즘은 여러 개의 에이전트가 비동기적으로 학습한다(그림 2-1 참조). 하나의 메인 쓰레드로부터 파생된 여러 개의 각 쓰레드가 독립적인 에이전트 역할을 수행한다. 메인 쓰레드는 전역 신경망을 가지고 있으며 전역 신경망은 정책망과 가치망으로 분리된다. 각각의 에이전트도 전역 신경망과 동일한 구조를 갖는 지역 신경망을 지닌다. 메인 쓰레드의 전역 신경망은 정책망 파라미터 벡터 θ 와 가치망 파라미터 벡터 θ_v 을 갖고, 에이전트의 지역 신경망은 정책망 파라미터 벡터 θ' 와 가치망 파라미터 벡터 θ'_v 를 갖는다.

정책망은 임의의 스텝 t 에서 상태 s_t 에 대한 행동 a_t 결정에 관한 정책 $\pi(a_t|s_t)$ 을 근사하고, 가치망은 상태 s_t 에 대한 기대보상을 나타내는 가치 함수 $V(s_t)$ 를 근사한다. 에이전트는 각 에피소드 마지막 스텝의 행동을 수행한 이후, 지역 신경망에 존재하는 정책망과 가치망을 통해서 메인 쓰레드가 가진 전역 신경망을 업데이트 한다. 이 후 새로운 에피소드의 학습이 수행될 때, 에이전트는 전역 신경망의 파라미터를 지역 신경망으로 복사하여 사용한다.

한편, Advantage-Function $A(s_t, a_t)$ 은 현재 정책에 의해 계산된 보상과

가치함수 $V(s_t)$ 와의 차이를 나타내는 지표이며 정책망 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 γ 를 활용하여 감쇠된 누적 보상에서 실제 가치 $V(s_t)$ 를 빼주는 방법을 활용함을 알 수 있다.

지역 신경망 내 정책망과 가치망의 Loss-Function은 다음과 같고,

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

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

정책망과 가치망의 파라미터 벡터의 업데이트 수식은 다음과 같다 [8].

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

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

Cross-Entropy의 MLE (Maximum Likelihood Estimation) 이론에 따르면 미래의 결과를 알 수 없는 상황에서는 정책망의 보상을 이용해서 불확실도를 낮추는 정책을 찾는다[15, 16]. 타깃 One-Hot 벡터 d_n 을 $[d_{n1}, d_{n2}, \dots, d_{nK}]^T$ 로 정의하고 y_k 를 k 번째 행동의 확률로 정의할 때 MLE $L(\theta)$ 의 수식은 다음과 같다 [16].

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

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.

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

제3절. 포트폴리오 투자 관리

금융자산 포트폴리오 관리 문제는 소유한 자산을 투자 가능한 자산들에 지속적인 분산투자를 하여 초기 투자 자산에 대비해 더 많은 자산을 얻는 것을 목표로 한다. 이와 관련된 선행 연구로서는 SCRP (Successive Constant Rebalanced Portfolios) [17], ONS (Online Newton Step algorithm) [18] 등이 있다. 이러한 연구들에서는 Price-Relative Vector를 정의하여 사용한다. $v_{i,t}$ 를 자산 i 에 대한 스텝 t 에서의 증가로 정의하고 v_t 를 전체 m 개의 자산에 대한 스텝 t 에서의 증가 벡터라고 정의할 때, 스텝 t 에서의 Price-Relative Vector y_t 는 시간 t 에서의 v_t 에 대해 이전 시간의 v_{t-1} 로 나누어 준 것으로 다음 수식과 같다 [10].

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

위 Equation (15)에서 y_1 은 $[1.0, 1.0, 1.0, \dots, 1.0]^T$ 로 가정하며, 본 논문에서도 Equation (15)에서 제시된 Price-Relative Vectors는 포트폴리오 관리를 위한 강화학습에서 활용한다. 하지만, 기존 연구 [17], [18]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 [19]을 비롯하여 딥러닝의 인공신경망을 활용하여 금융지표 중 하나인 S&P 500을 예측 하려는 연구인 [20]도 포트폴리오 관리를 직접 수행하는 방안을 제시하지 않는다.

한편, [21, 22, 23]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 에이

전트는 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 [10]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 활용하여 [10]에서 제시된 DPG와 접목하여 기존 방법보다 분산 투자 수익률이 우수하다는 결과를 비교 평가를 통해 제시하였다.

제3장 데이터 처리

본 논문에서 기준 통화는 원화로 정하며, 국내 임의의 거래소에서 원화 기반의 금융자산 가격 데이터를 취득 및 관리한다. 임의의 거래소에서 거래가 가능한 자산의 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.

<표 3-1> 수집된 BTC 데이터에 존재하는 이상 데이터 예시

	Timestamp	Date&Time	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

제1절. 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(표 3-1의 5, 6번째 행). 마지막으로 제로 데이터는 단위 시간 T 간격으로 수집되었으나 가격 정보가 '0'으로 기록되어있는 경우이다(표 3-1의 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. 표 3-1은 예시로서, 단위 시간 T 가 10분 일 때 암호화폐의 기축통화로 통용되는 BTC (Bitcoin)에서 발생한 비정상 데이터의 예시이다. 아래 1)~3)에서는 표 3-1을 예로 들어 비정상 데이터에 대한 처리 방법에 대해 설명한다.

1. 부적합 데이터 처리

매 10분 정각마다 금융자산 데이터를 수집할 때, API를 통해 가져오는 데이터의 타임스탬프가 10분 정각의 데이터가 아닐 수가 있다. 이러한 경우, 단순히 단위 시간 10분 기준으로 잘못된 타임스탬프 값을 수정한다. 표 3-1의 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 (표 3-1의 Timestamp도 이에 따라 함께 변경한다).

2. 손실 및 제로 데이터 처리

수집된 임의의 데이터 내에 연속된 k 개의 가격 정보가 없거나 0인 경우에, 바로 이 전 가장 가까운 올바른 데이터 정보를 X_a 라고 하고 바로 이

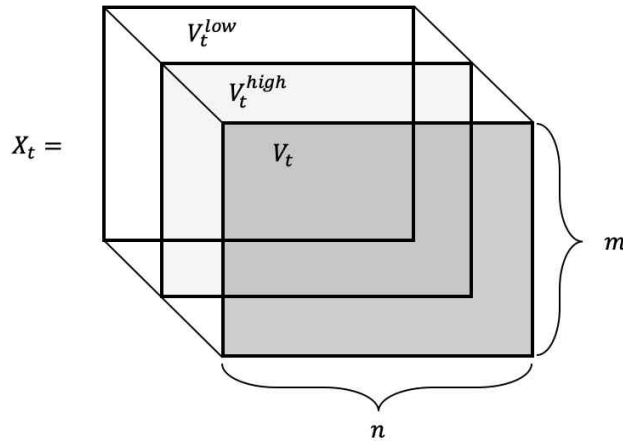
후 가장 가까운 올바른 데이터의 정보를 X_{a+k+1} 라고 할 때, 유실된 가격 정보 X_{a+i} ($1 \leq i \leq k$)는 다음 수식과 같이 보정된다.

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

손실 데이터와 제로 데이터의 전처리 방법은 같고, 제로 데이터의 수정 방법을 예로 들면, 표 3-1의 2, 3번째 행의 증가는 연속해서 제로 데이터가 수집되었다. 이 경우 Equation (16)을 활용하여 데이터를 수정한다. X_a 는 7,200,000이고, k 는 2이고 X_{a+3} 은 7,206,000이다. X_{a+1} 인 2번째 행의 증가는 0에서 7,202,000으로, X_{a+2} 인 3번째 행의 증가는 0에서 7,204,000으로 치환된다.

제2절. 데이터셋 정의

비정상 데이터가 앞서 설명된 전처리 방법으로 처리된 이후에는 금융자산의 시가를 제외한 종가, 고가, 저가 데이터를 활용하여 데이터셋이 구성된다. 금융자산 가격 데이터셋은 시계열 데이터이며, 단위 시간 T 를 기준으로 형성되는 임의의 시점 t 의 종가가 대부분의 경우 시점 $t+1$ 의 시가와 동일하기 때문에 데이터셋에서 시가를 제외하였다. 총 m 개의 자산 종류에 대하여, T 를 기준으로 과거 n 개의 종가, 고가, 저가 데이터를 활용하여 데이터셋을 구성한다. 이 때, n 은 윈도우 크기(Window Size)라고 일컫는다. 따라서 임의의 시점 t 의 데이터셋 X_t 의 차원 구성(shape)은 $(3, m, n)$ 이다. 이후 5장에서 알 수 있듯이 본 논문에서 단위 시간 T 는 기본적으로 10분으로 정한다.



[그림 3-1] 윈도우 크기 n 에 대한 데이터셋 X_t 의 구조

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

$$v_{0,t} = v_{0,t}^{high} = v_{0,t}^{low} = 1 \quad (17)$$

또한, V_t , V_t^{high} , V_t^{low} 벡터는 m 개의 자산에 대한 임의의 시점 t 를 기준으로 정규화된 종가, 고가, 저가 벡터이다. 각각의 벡터에 대하여, 윈도우 크기 n 에 대한 마지막 종가 대비 벡터 요소별 나누기 연산(\oslash)을 통해 벡터 내에 존재하는 기존 가격 데이터를 다음과 같이 정규화한다.

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

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

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

그림 4-1은 임의의 시점 t 에서의 X_t 를 도식화한 것이다.

제4장 강화학습 모델

제1절. 보상정의

본 논문에서 제안하는 포트폴리오 관리 대상 자산의 종류는 사전에 m 개를 미리 정한다고 가정한다. 본 모델에서 가정하는 시간은 시점 0부터 시작하며 금융자산 분배 행동은 단위 시간 T 를 기준으로 형성되는 매 시점 t ($t \geq 1$)마다 강화학습의 스텝 t 가 수행된다. 임의의 스텝 t 에서의 가중치 벡터(Weight Vector) w_t 는 각 자산별 자산 분배율 벡터로 정의되며, 이 벡터를 구성하는 요소 $w_{i,t}$ 는 스텝 t 에서의 자산 i 의 자산 분배율이다 ($0 \leq w_{i,t} \leq 1$). $w_{0,t}$ 는 스텝 t 에서 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 t 에 대해 기준 통화를 포함하여 각 자산 분배율의 총합은 항상 1이다.

$$\sum_{i=0}^m w_{i,t} = 1 \quad (21)$$

에이전트는 자산 분배율에 따라 자산을 매매하며, 스텝 $t-1$ 에서 m 개의 자산에 대한 종가 벡터 v_{t-1} 가 스텝 t 에서 v_t 로 변동됨에 따라서 각 자산의 잔고가 변경된다. 각 자산의 보유 잔고는 스텝 t 에서 자산 m 개에 대한 보유 수량을 의미하며 b_t 로 표현된다.

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

$$p'_{t-1} = v_t \cdot b_{t-1} \quad (22)$$

한편 자산 분배 시에 따른 수수료가 지불되기 때문에, 자산 분배 전 변동 포트폴리오 자산 가치에서 분배되는 자산의 분배량에 따라 수수료가 지불된다. 거래되는 자산의 양에 따라 수수료비율 c 가 적용되어 수수료가 계산된다. 스텝 t 에서 수수료가 지불된 이후 포트폴리오 자산 가치 p_t 와 보유 잔고 b_t 는 다음과 같이 정의된다(연산자 \odot 는 두 벡터의 동일 위치 원소들의 곱셈을 의미한다).

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

$$b_t = p_t w_t \odot v_t \quad (24)$$

한편, 스텝 t 에서 로그 손익률 r_t 는 다음과 같이 표현 가능하다.

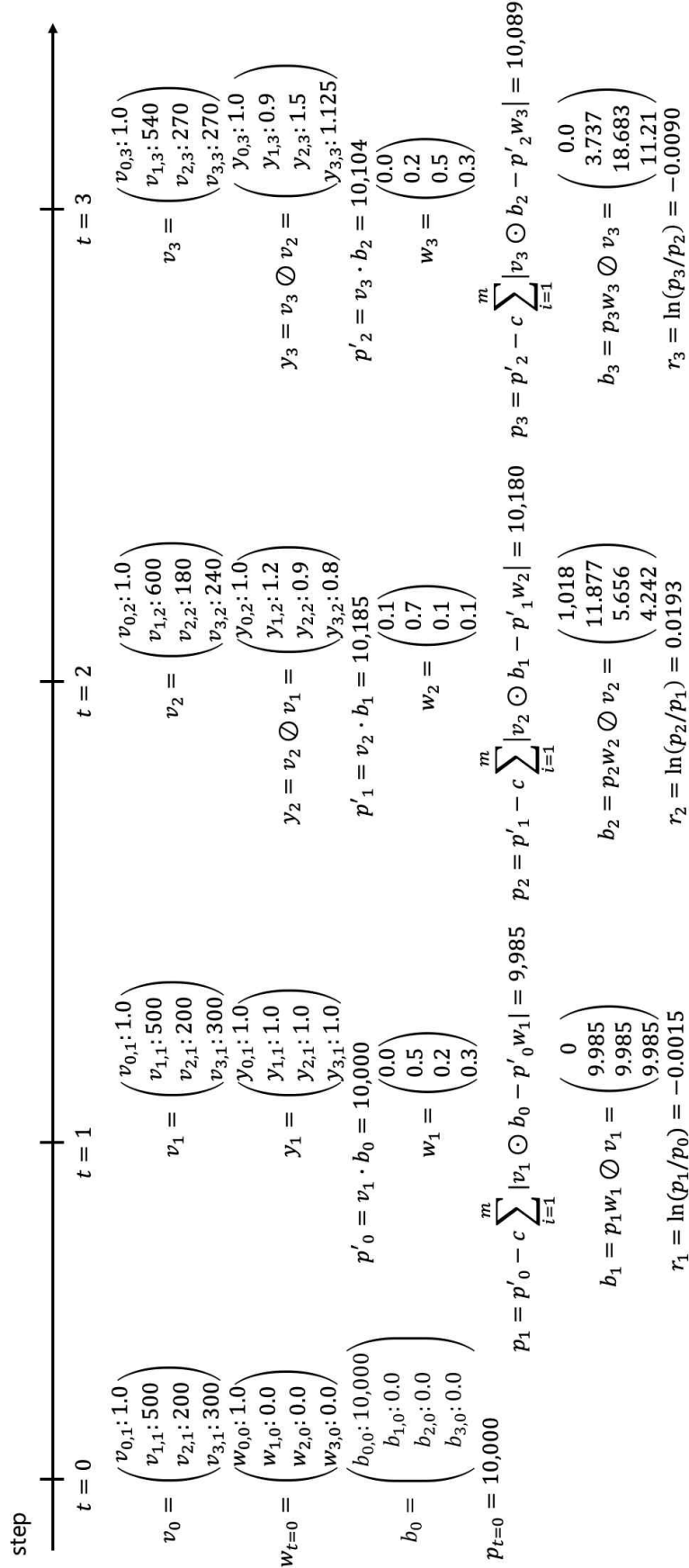
$$r_t = \ln(p_t/p_{t-1}) \quad (25)$$

강화학습 에이전트가 스텝 t 에서 자산을 분배시키는 행동을 한 후에 받는 보상이 바로 Equation (25)에 기술된 r_t 가 된다.

그림 4-1은 임의로 선정한 금융자산 3종에 대해서 시점 0 이후 3번의 스텝에 걸쳐서 자산을 분배시키는 에이전트의 보상 r_t 의 산출 과정을 상세한 예시로서 보여준다. 이 예에서 수수료 비율 c 는 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_t 는 -0.0015이다. 이 후 스텝 $t=2$ 에서 포트폴리오 자산 가치는 10,721로 증대되었고, 로그 손익률 r_t 는 0.0711로 긍정적인 보상이 나왔다. 마지막 스텝 $t=3$ 에서는 포트폴리오 자산 가치가 10,907로 소폭 증대되었고, 로그 손익률 r_t 은 0.0172로 계산되었다.

$m = 3$ c (commission ratio) = 0.0015

(i | CASH: 0, A asset: 1, B asset: 2, C asset: 3)



[그림 4-1] 포트폴리오 관리 과정 예시

제2절. Mini-Batch Sampling

강화학습 에이전트의 궁극적인 목표는 마지막 스텝 t_f 에서 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[24]. 샘플링 잡음(Sampling-Noise)과 샘플링 편향(Sampling-Bias) 문제를 피하기 위해서, 미니배치 샘플링 방식에 다음 수식과 같은 기하 분포 확률(Geometrically Distributed Probability) $P_\beta(b)$ 를 이용하여 임의의 스텝 t 에서의 샘플 데이터의 시작 스텝 b 를 정하였다.

$$P_\beta(b) = \beta(1 - \beta)^{t-b-N} \quad (26)$$

즉, 미니배치 샘플링을 할 때에 $P_\beta(b)$ 에 근사하여 $b \leq t - N$ 를 만족하면서 무작위로 선택된 b 및 미리 정의한 미니배치 크기 N 에 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.

제3절. Deterministic Policy Gradient based A3C Model

강화학습 에이전트의 궁극적인 목표는 마지막 스텝 t_f 에서 최종 포트폴리오 자산 가치를 최대화 시키는 것이다. 시계열 데이터의 특성상 입력 데이터의 수는

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

대부분의 강화학습은 신경망 모델을 거쳐 나온 결과 값 중에서 가장 값이 큰 행동을 선택한다. 하지만 본 논문의 에이전트는 매 스텝마다 각 원소의 총 합이 1인 확률 벡터 $[p[a_1|s, \theta], p[a_2|s, \theta], \dots, p[a_n|s, \theta]]$ 자체를 에이전트 행동, 즉 4.1절에서 정의한 임의의 스텝에서의 w_t 로 활용한다. 따라서 모델을 거쳐 나온 행동을 Cross-Entropy 수식 Equation (13)의 인자 d_n 과 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 $[p[a_1|s, \theta], p[a_2|s, \theta], \dots, p[a_n|s, \theta]]$ 의 원소 각각의 제곱을 모두 더한 값을 대입하였다. 정책망 모델을 통하여 얻은 행동 벡터의 k 번째 원소 y_k 를 $p[a_k|s, \theta]$ 로 정의할 때, 변형된 MLE $L'(\theta)$ 및 Cross-Entropy $E'(\theta)$ 을 다시 정리하면 다음과 같다.

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

$$E'(\theta) = -\log L'(\theta) \quad (28)$$

에이전트에게 주어진 보상 체계는 Full-Exploitation 방식으로 정하여 항상 정책망에서 제시하는 포트폴리오 분배 가중치에 따라 자산을 분배한다. 매 스텝마다의 상태와 행동을 모두 고려하는 보상 함수를 정책 매개변수 업데이트에 사용하며, 스텝 f 가 종료된 시점의 에피소드의 최종 보

상 함수 R_f 는 다음과 같다.

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

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

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

정책망과 가치망의 업데이트 수식은 다음과 같다.

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

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

제4절. 실제 시장과 투자 시뮬레이션과의 차이

실제 금융자산 거래시장에서는 유망한 자산으로 사람들의 투자가 몰리며, 반대로 유망하지 못한 자산의 거래 규모는 매우 작다. 또한, 특정 매수/매도자에 의하여 자산의 가치가 쉽게 변동될 수 있다. 이러한 상황은 시장 영향(Market Impact)이라고 언급되며, 실제 시장이 아니라 본 논문에서처럼 이전 가격 데이터를 기반으로 투자 시뮬레이션을 하는 환경에서는 그러한 시장 영향 현상이 없다고 가정한다.

또한, 현 시점에서 매매되어야 하는 종의 매도 수량 혹은 매수 수량이 없다면, 혹은 매도/매수자의 호가가 현저히 높거나 낮다면 매매자로서는 손해를 감수할 수밖에 없다. 실제 시장에서 시장가주문 시점의 가격으로 체결되지 않는 경우가 빈번히 생길 수밖에 없는 이유이다. 강화학습 에이전트가 동작하는 투자 시뮬레이션에서는 이러한 체결오차(Slippage)는 없다고 가정한다.

제5장 실험 평가

금융자산 포트폴리오 관리 목적은 보유한 통화를 거래 가능한 다른 자산과 교환하여 최종 포트폴리오 자산 가치를 증대시키는 것이다. 실험 평가에서는 1) 본 논문이 제시하는 모델인 A3C-DPG, 2) [10]의 연구에서 제시된 DPG, 그리고 3) 무작위로 포트폴리오 관리를 하는 Random 모델에 대한 백테스트(Back-Test)를 진행한다. 그리고 국내 빗썸 거래소의 API [25]를 활용해 취득한 거래 가능한 모든 암호화폐를 금융자산으로써 활용했다.

<표 5-1> 백테스트를 위한 데이터의 기간

시장 상황	ID	T	Learning data period	Back-Test data period
일 반	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
상승장	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
하락장	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

제1절. 백테스트 및 강화학습 성능 평가 기준

백테스트가 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 t 에서 PVVR P^t 의 수식은 아래와 같으며,

$$P^t = p_t / p_{t-1} \quad (33)$$

마지막 스텝 f 에서 누적된 PVVR P^f 은 다음과 같다.

$$P^f = \prod_{t=1}^f P^t = p_f / p_0 \quad (34)$$

한편 모델이 가변적인 시장 상황에 따라 유연하게 대응할 수 있는지를 평가하기 위해서 백테스트의 기간을 ‘일반’, ‘상승장’, ‘하락장’으로 나누어 비교하였다. 그리고 단위 시간 T 를 10분과 30분으로 나누어 자산의 분배 주기에 따른 모델 성능을 비교평가 하였다. 3.2절에서 설명한 임의의 데이터셋 샘플의 윈도우 크기 n 은 단위 시간 기준으로 100으로 정하였다. 따라서 강화학습 모델에 입력으로 사용되는 임의의 데이터셋 샘플 1개의 시간 길이는 T 가 10이면 1000분(=16시간 40분)이며, T 가 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 빚썸 거래소에 상장되고 거래량도 많은 암호화폐들이다. 표 5-1은 학습에 사용된 학습 데이터 기간과 백테스트를 위한 데이터 기간을 제시한다. 각 모델들의 학습시, A3C-DPG 모델은 8개의 에이전트가 각각 5,000번의 에피소드가 수행하고, DPG 모델은 단일 에이전트가 50,000번의 에피소드를 학습한다. 학습이 진행되는 각 에피소드 마다 미니배치 샘플링 되는 전체 데이터 기간은 2일이다. 그래서 T 가 10이면 미니배치 샘플링 크기 N 은 288개(=6×24(Hour)×2(Day))이고, T 가 30이면 미니배치 샘플링 크기 N 은 96개(=2×24(Hour)×2(Day))이다. 학습이 종료되면 에이전트는 학습 데이터 기간 직후 바로 이어지는 15일 간의 백테스트를 진행하며, 백테스트 종료 후에는 누적된 PVVR P^f 가 산출된다. 백테스트가 시작되면 에이전

트는 학습과 동일한 방식으로, T 만큼의 시간이 지나는 매 스텝 t 마다 포트폴리오 재분배를 하게 된다. 백테스트가 종료된 후 산출된 최종 PVVR P^f 를 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 [26, 27]. 추적오차는 백테스트 기간 내 각 스텝 t 마다의 포트폴리오 가치 증감 비율인 $P_{A3C-DPG}^t$ 과 P_{DPG}^t 를 비교함으로써 DPG와 A3C-DPG 모델 간의 성능격차를 측정한다. 다음은 실험에 사용된 추적오차 계산 수식이다.

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

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 $P_{A3C-DPG}^f - P_{DPG}^f$ 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.

$$IR = \frac{P_{A3C-DPG}^f - P_{DPG}^f}{TE} \quad (36)$$

제2절. 모델 성능 비교 결과

<표 5-2> A3C-DPG, DPG, Random 모델 간 최종 PVVR 비교

	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

<표 5-3> A3C-DPG와 DPG 모델 간 추적오차와 정보비율

	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

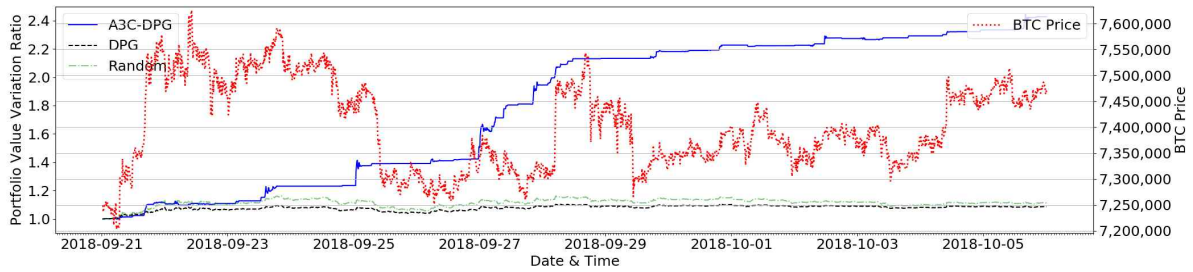
그림 5-1~5-6은 3가지 백테스트 기간 동안 각 모델의 포트폴리오 가치 증감 비율인 PVVR의 변화 추이를 암호화폐들의 기축통화로 통용되는 BTC의 증가와 함께 그래프로 보여주며, 표 5-2는 최종 PVVR 만을 각 모델별로 제시한다. 그림 5-1, 5-2는 거래 단위 시간 T 가 10분과 30분에 대하여 가격변동이 크지 않은 일반적인 시장상황인 Back-Test #1-1, #1-2 동안의 PVVR 변화 추이를 보여준다. 한편, 가격 변동이 상승세를 타며 투자하기 좋은 시장상황을 나타내는 Back-Test #2-1, #2-2에 대하여, 그림 5-3, 5-4는 단위 시간 T 가 각각 10분과 30분일 때의 PVVR 변화 추이를 보여준다. 마지막으로, 가격 변동이 하락세를 타며 투자하기 좋지 못한 시장상황을 나타내는 Back-Test #3-1, #3-2에 대하여, 그림 5-5, 5-6은 단위 시간 T 를 각각 10분과 30분일 때의 PVVR 변화 추이를 보여준다.

제시된 표 5-3에서 보이듯이, 본 논문에서 제안하는 A3C-DPG 모델이

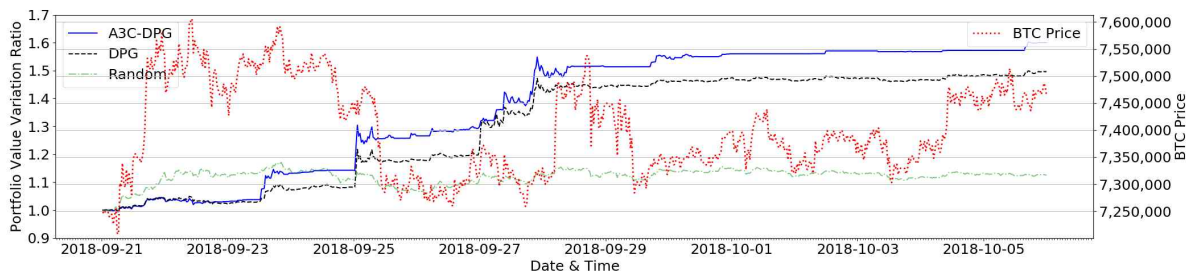
Random 모델뿐만 아니라 기존 모델인 DPG 보다 전체적으로 더 높은 최종 PVVR을 산출함을 알 수 있다. 또한, 제안하는 A3C-DPG 모델인 경우 T 가 30분인 경우 보다는 10분인 경우에 최종 PVVR이 높다. 즉, A3C-DPG 모델은 짧은 주기의 투자 운용을 수행하여 시장 상황에 더 민첩하게 대응할 수 있음을 암시한다. 또한, 가격 변동이 상승세를 타며 투자하기 좋은 시장상황에서는 T 가 10분일 때 A3C-DPG 모델의 최종 PVVR은 17.3247로서 최초 자산 대비 17배 이상의 많은 수익을 거두었음을 알 수 있다.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

그리고 백테스트시 강화학습 에이전트의 PVVR 추이를 그림 5-1~5-6에서 보이고 있다. PVVR 곡선이 대체적으로 계단형태를 보이는 그림 5-1, 5-2는 기축통화인 BTC 가격이 상승세를 띄면 PVVR 곡선이 가파르게 증가하는 추세를 보이고, BTC 가격이 하락하면서부터는 강화학습 에이전트가 분산 투자를 멈추고 자산을 유지하는 형태를 나타낸다. 같은 계단 곡선을 보이는 그림 5-5, 5-6은 BTC의 가격이 하락세가 지속될 때 투자를 중단하여 PVVR 곡선이 수평을 유지하는 폭이 긴 것을 볼 수 있다. 반면, 그림 5-3, 5-4와 같이 BTC 가격이 지속적인 상승세를 보이면 PVVR 곡선도 마찬가지로 지속적으로 증대되는 모습을 보인다.

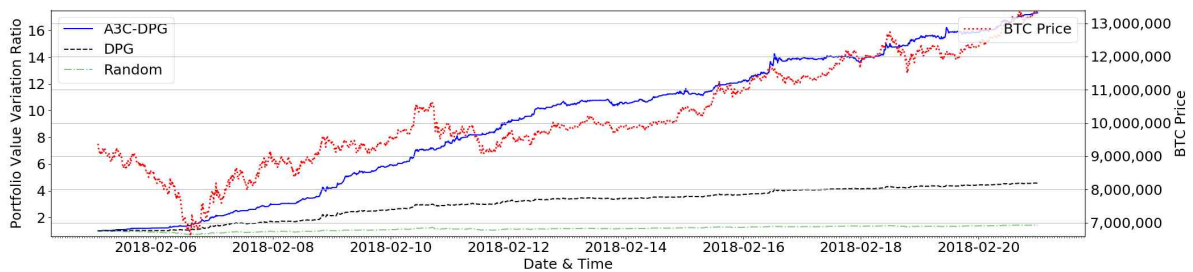
또한 백테스트에서 A3C-DPG 모델이 DPG 모델보다 더욱 활발한 투자 운용을 했음을 표 5-3이 제시하는 추적오차와 정보비율 지표를 통해서 알 수 있다. Back-Test #2-1의 경우 A3C-DPG 모델이 DPG 모델보다 높은 초과 투자 수익을 얻었기 때문에 7.2513이라는 높은 추적오차가 발생했음에도 정보비율이 1.7588로 계산되었음을 알 수 있다. 반면 Back-Test #3-1은 A3C-DPG 모델이 DPG 모델보다 효율적인 투자 움직임을 보였기 때문에 0.2431의 낮은 추적오차에도 불구하고 2.4596이라는 높은 정보비율이 계산되었음을 확인할 수 있다.



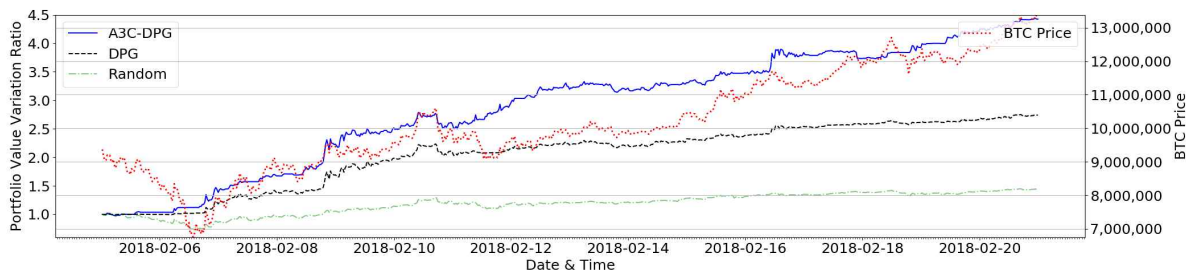
[그림 5-1] Back-Test #1-1: 일반적인 시장 상황에서의 포트폴리오 관리,
 $T = 10$ 분, (2018-09-21 00/10/00 ~ 2018-10-06 00/00/00)



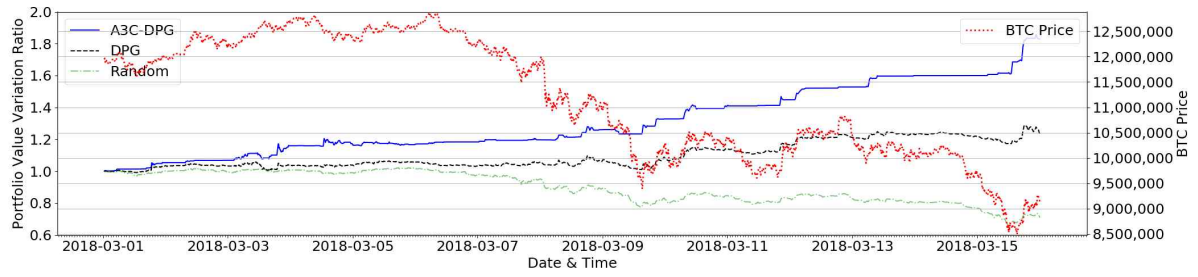
[그림 5-2] Back-Test #1-2: 일반적인 시장 상황에서의 포트폴리오 관리,
 $T = 30$ 분, (2018-09-21 00/30/00 ~ 2018-10-06 00/00/00)



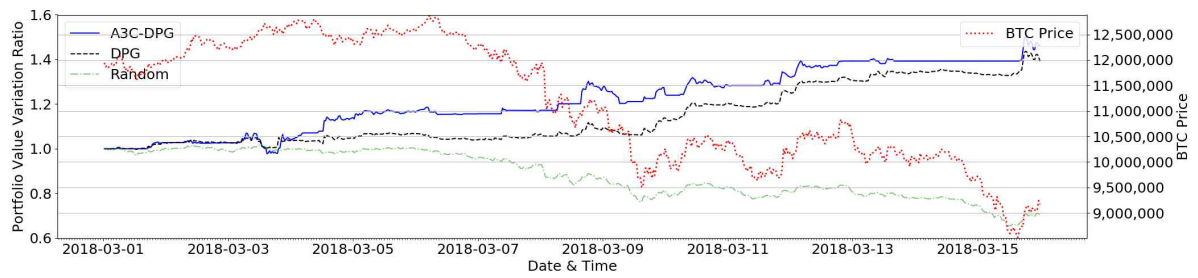
[그림 5-3] Back-Test #2-1: 상승장에서의 포트폴리오 관리,
 $T = 10$ 분, (2018-02-05 00/10/00 ~ 2018-02-21 00/00/00)



[그림 5-4] Back-Test #2-2: 상승장에서의 포트폴리오 관리,
 $T = 30$ 분, (2018-02-05 00/30/00 ~ 2018-02-21 00/00/00)



[그림 5-5] Back-Test #3-1: 하락장에서의 포트폴리오 관리,
 $T = 10$ 분, (2018-03-01 00/10/00 ~ 2018-03-16 00/00/00)

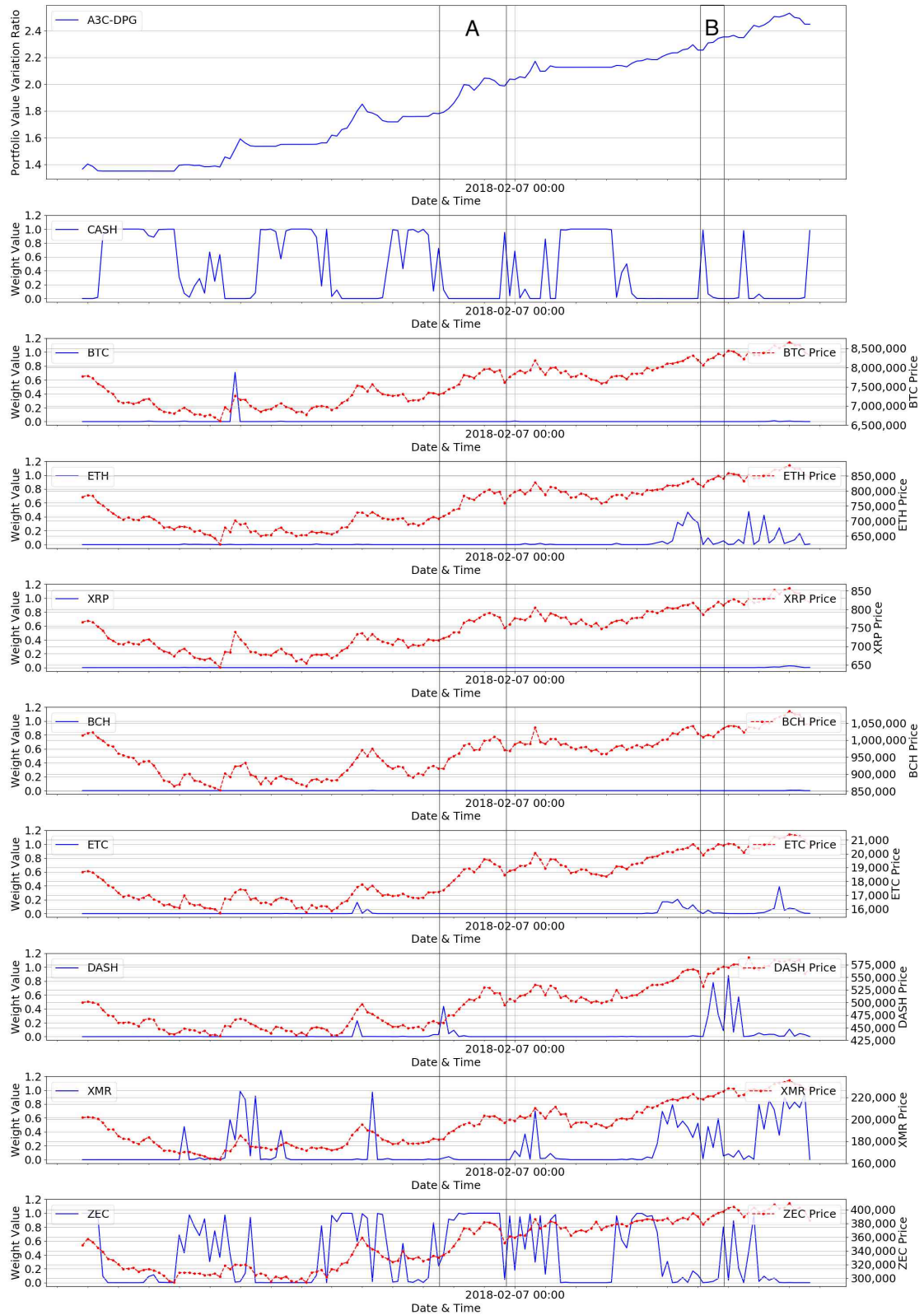


[그림 5-6] Back-Test #3-2: 하락장에서의 포트폴리오 관리,
 $T = 30$ 분, (2018-03-01 00/10/00 ~ 2018-03-16 00/00/00)

제3절. 자산 분배율 분석

이번 절에서는 에이전트가 백테스트에서 높은 FPV를 얻을 수 있었던 이유를 그림 5-7에서 제시하는 암호화폐 자산 분배율 분석을 통해서 알아본다. 그림 5-7은 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.



[그림 5-7] 백테스트에서의 자산 분배 비율

제6장 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 수집하여 금융자산 포트폴리오 관리 환경의 데이터로 사용했고, 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 구매받지 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 이루어진다면 강화학습을 금융 투자 도메인에서도 활용할 수 있는 가능성이 존재한다. 한편 시장 영향과 체결오차 문제는 포트폴리오 관리에 있어 주요한 난제이며 향후 연구를 통해 그러한 문제를 해결할 계획이다.

참 고 문 헌

- [1] “GUNBOT - Crypto Trading Bot,” GUNBOT, <https://www.gunbot.com>, 2018.
- [2] “start [ProfitTrailer Wiki],” ProfitTrailer, <https://wiki.profittrailer.com/doku.php?id=start>, 2018.
- [3] I. Kaastra, M. Boyd, “Designing a neural network for forecasting financial and economic time series,” *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996.
- [4] Candela, “Dataset shift in machine learning,” London: MIT Press, 006.3 CAN, 2009.
- [5] Kim YB, “Predicting Fluctuations in Cryptocurrency Transactions Based on User Comments and Replies,” *PLoS ONE* 11(8): e0161197, 2016.
- [6] S. McNally, “Predicting the Price of Bitcoin Using Machine Learning,” 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing, pp. 339–343, March, 2018.
- [7] Sutton, R. and Barto, A. “Reinforcement Learning: an Introduction,” MIT Press, 1998.
- [8] Volodymyr Mnih, “Asynchronous Methods for Deep Reinforcement Learning,” *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, JMLR: W&CP, vol. 48, 2016.
- [9] A. Nair, “Massively Parallel Methods for Deep Reinforcement Learning,” *Deep Learning Workshop, International Conference on Machine Learning*, Lille, France, 2015.
- [10] Zhengyao Jiang, “A Deep Reinforcement Learning Framework for

- theFinancial Portfolio Management Problem,” JMLR, 30 pages, 5 figures, 2017.
- [11] Christopher JCH Watkins and Peter Dayan. “Q-Learning,” Machine Learning, 8(3-4):279-292, 1992.
 - [12] Kai Arulkumaran, “A Brief Survey of Deep Reinforcement Learning,” in IEEE Signal Processing Magazine Special Issue On Deep Learning For Image Understanding, 2017.
 - [13] Hado van Hasselt, “Deep Reinforcement Learning with Double Q-learning,” Proceedings of 30th AAAI Conference on Artificial Intelligence (AAAI-16)
 - [14] 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, pages 387-395, 2014.
 - [15] K. Chepuri, T. Homem de Mello, “Solving the vehicle routing problem with stochastic demands using the cross entropy method,” Annals of Operations Research, 2004.
 - [16] G. Alon, D. P. Kroese, T. Raviv, R. Y. Rubinstein, “Application of the Cross-entropy method to the buffer allocation problem in a simulation-based environment,” Annals of Operations Research, 2004.
 - [17] Gaivoronski, “Stochastic nonstationary optimization for finding universal portfolios,” in Annals of Operations Research 100(1), 165 - 188 (2000)
 - [18] Agarwal, A., “Algorithms for portfolio management basedon the newton method,” in ICML, New York, NY, USA (2006)
 - [19] Bin Li, Peilin Zhao, Steven C. H. Hoi, and Vivekanand Gopalkrishnan. “Passive aggressive mean reversion strategy for portfolio selection,” PSML, Machine Learning, 87(2):221-258, 2012.
 - [20] 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*, 9(1):1, 2013.
- [21] Katia Sycara, K. Decker and Dajun Zeng, “Designing a Multi-Agent Portfolio Management System,” *Proceedings of the AAAI Workshop on Internet Information Systems*, 1995.
- [22] K. Sycara, A. Pannu, M. Williamson, Dajun Zeng, K. Decker, “Distributed intelligent agents,” *IEEE Expert*, Volume 11, Issue 6, Dec 1996.
- [23] 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.
- [24] Mnih, Volodymyr, “Human-level control through deep reinforcement learning,” in *Nature*, 518(7540):529 - 533, 02 2015.
- [25] “API - Bithumb,” Bithumb, <https://www.bithumb.com/u1/US127>, 2018.
- [26] Mu Li, “Efficient Mini-batch Training for Stochastic Optimization,” In 2014 ACM, 978-1-4503-2956-9, 2014.
- [27] Edward Qian, “Active Risk And Information Ratio,” *JOIM*, Vol. 2, No. 3, (2004), pp. 1 - 15.

ABSTRACT

Financial Portfolio Management Based on Asynchronous Deep Reinforcement Learning

Kim, Ju bong

Department of Computer Science & Engineering

Major of Computer Science & Engineering

The Graduate School

Korea University of Technology Education

In the financial investment management strategy, dividing the investor's assets into several financial products is called portfolio management, and a myriad of assets in the financial market make an efficient investment by creating a portfolio. 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 the Asynchronous Advantage Actor-Critic (A3C) based model, which is a representative asynchronous deep reinforcement learning algorithm, based on the existing deep reinforced learning-based financial portfolio investment strategy. In addition, since the conventional cross-entropy function can not be applied as it is in the course of incorporating A3C, the existing cross-entropy is modified to fit the portfolio investment method and the solution is presented. Finally, we compare with the existing

reinforcement learning based cryptography portfolio investment algorithm, and proved that the performance of A3C model proposed in this paper is excellent.

Keyword : Financial Portfolio Management, Reinforcement Learning, Distributed Investment, Asynchronous Advantage Actor-Critic, Cryptocurrency