Journal of the Korean Institute of Industrial Engineers
Vol. 47, No. 4, pp. 399-405, August 2021.
ISSN 1225-0988 | EISSN 2234-6457
https://doi.org/10.7232/JKIIE.2021.47.4.399

<Application Research>

강화학습 기반 주식 자동 매매 모델 전략 제안

황호현[1] · 김용훈[2] · 이영훈[2†]

[1]서울과학기술대학교 데이터사이언스학과 / [2]서울과학기술대학교 산업공학과

Suggestion of Strategy for the Automation of Stocks Based on Reinforcement Learning

Hohyun Hwang[1] · YongHoon Kim[2] · YoungHoon Lee[2]

[1]Department of Data Science Seoul National University of Science and Technology
[2]Department of Industrial Engineering Seoul National University of Science and Technology

This work proposes an effective learning strategy by developing a stock auto-sell system based on reinforcement learning and comparing and analyzing it under various conditions. Based on PG, DQN, and A2C reinforcement learning techniques, reinforcement learning was implemented by determining the behavior of buying/selling/viewing networks at the end of the day and providing compensation. The experiment was conducted largely by dividing four conditions. We propose an effective reinforcement learning strategy through performance comparison by reinforcement learning technique, model stability comparison by variability in learning period, and performance comparison experiment by length of learning period of model.

Keywords: Reinforcement Learning, DQN, PG, A2C, Stock Forecasting

1. 서 론

머신러닝을 통한 주가 예측은 금융 분야에서 매우 펀더멘탈한 연구로써 다양한 알고리즘에 기반한 연구들이 이루어지고 있다. Shen *et al.*(2012)은 SVM과 트리 기반 모델인 MART를 이용하여 주가 예측 모델을 제안하였고, 둘의 성능을 비교하는 연구를 진행했다. Adebiyi *et al.*(2014)은 시계열 예측에 사용되는 고전적인 ARIMA기법을 통한 예측으로 주가를 예측한 연구를 진행하고, 이어서 ARIMA를 이용한 예측 연구와 인공신경망을 통해서 주가를 예측한 연구를 비교하여 신경망의 성능을 비교하고 신경망의 주가 예측 성능이 우수함을 설명했다. Chen *et al.*(2017)은 일반적으로 시계열 예측기법으로 사용되지 않았던 합성곱 신경망을 이용해 주가지수 등락을 예측하는 연구를 진행했다. Lee(2017)은 인공 신경망이 주가예측에 효과적이라는 결과를 바탕으로 여러 종류의 신경망을 사용하여 성능을 비교했다. Singh *et al.*(2017)은 여러 종류의 인공 신경망에 PCA를 통해 나타난 특징을 결합하는 방법을 제안함으로써 기존에 신경망만 이용했을 때 보다 더 나은 모델을 제안한다. Won *et al.*(2018)은 기술적 분석지표를 주가 데이터와 결합하고, 딥러닝을 활용하여 주가를 예측했다. 그런데 대부분의 연구들이 특정 일자를 기준으로 예측을 진행하기 때문에 실시간으로 즉각적인 의사 결정이 요구되는 자동매매 시스템이 직접적으로 응용하는 데에는 한계를 가지고 있다. 따라서 즉각적으로 이루어지는 매도/매수 의사결정을 고려했을 때, 보상과 행동을 모델링할 수 있는 강화학습이 효과적인 대안이 될 수 있다. Deng(2017)는 논문에서 딥러닝 인공신경망 이용해서 정보기능을 학습하고, 그에 맞는 행동과 보상을 판단하는 방법을 강화학습을 적용하였다. Lee (2001)은 강화학습 알고리즘 TD(0)를 적용하여 인공 신경망에 의한 함수 근사치를 구하여 주어진 시간에 알맞은 주가 상태를 학습하였다. Lee *et al.*(2002)는 TD 알고리즘과

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2020R1F1A1067914).
† 연락저자 : 이영훈 교수, 01811 서울시 노원구 공릉로 232 서울과학기술대학교 산업공학과, Tel : 02-970-6467, Fax : 02-970-6467,
E-mail : yhoon.lee@seoultech.ac.kr
2021년 4월 14일 접수; 2021년 6월 10일 수정본 접수; 2021년 6월 21일 게재 확정.

Q 알고리즘을 사용하여 주가를 예측하는 R-Trader를 제안한다. Azhikodan(2019)는 정책 경사 기반의 강화학습 모델을 통해서 자동 거래 봇을 개발함으로써 강화학습이 주식 거래에 적합하다는 것을 증명하였다. 이렇게 실제로 많은 연구에서 강화 학습에 기반한 자동 주식 매매 시스템을 제안하고 있다.

다만 Won *et al.*(2002)은 기존 강화학습이 주식 분야에 적용된 연구들의 한계점을 종목이나 기간, 추세에 따라 그 효과가 매우 달라지는 것이라고 지적하고 있다. 본 연구에서는 강화학습에 기반한 주식 자동 매매 시스템을 구현하고, 종목이나 학습 기간 등을 달리 적용하여 실제 투자 시뮬레이션 결과를 산출, 비교하여 변동성이 큰 주가 데이터 분석에 대해서 효과적인 전략을 제안하고자 한다.

본 논문은 제 2장에서는 강화학습에 사용된 알고리즘을 소개하고, 제 3장에서는 본 연구에서 사용된 전체적인 모델에 대한 설명, 제 4장에서는 본 연구의 실험 과정 및 결과에 대해서 설명하고, 제 5장은 결론으로 구성되어 있다.

2. 관련 연구

강화학습은 어떤 환경(environment) 안에서 주어진 상태(state)에 따라 주체(agent)가 보상(reword)을 최대화하는 행동(action)을 배우는 학습법이다. 주체(agent)는 환경(environment)과 상호작용하며 행동(action)의 결과로 주어지는 보상(reword)을 통해 더 좋은 행동(action)을 학습한다. 본 논문에서는 Q-learning을 바탕으로 한 Deep Q Networks(DQN), Policy Gradient(PG), Actor-Critic을 발전시킨 Advantage Actor-Critic(A2C) 세 가지 기법을 사용한다.

2.1 Q-Learning

강화학습은 주변 상태(s)에 따라 어떤 행동(a)을 할지 판단을 내리는 주체인 에이전트가 있고, 에이전트가 속한 환경이 있다. 에이전트가 행동을 하게 되면 그에 따라 상태(s')가 바뀌게 되고, 보상(r)을 받을 수도 있다. 강화학습의 목표는 주어진 환경에서 보상을 최대한 많이 받을 수 있는 에이전트를 학습하는 것이다. 계속해서 행동과 관측을 하며 식 (1)과 같이 상태와 행동에 대한 가치함수인 Q 함수를 업데이트 한다.

$$Q(s,a)=r(s,a)+\gamma Q(s',a) \tag{1}$$

강화학습에는 항상 최선의 선택만을 고집하는 $greedy$ 알고리즘과 0과 1사이의 값을 가지는 ϵ를 이용해서 $p=1-\epsilon$ 의 확률로는 원래 했던 대로 최선의 행동을 선택하고, 나머지 $p=\epsilon$의 확률로는 임의로 행동을 취하게 하는 $\epsilon-greedy$ 알고리즘이 사용된다. 대부분의 경우 특정 상태 s에서 취할 행동 a는 Q 함수에 대해 $\epsilon-greedy$ 알고리즘으로 선택하는 반면, 업데이트 목표 값으로 쓰이는 행동 a는 $greedy$ 알고리즘을 사용한다.

$$Q(s,a)=r(s,a)+\gamma\max_{a} Q(s',a) \tag{2}$$

정리하면 상태 s에서, $\epsilon-greedy$를 통해 이번에 취할 행동 a를 고른 뒤, 실제로 시행해 바뀐상태 s'와 보상 r을 관측하고 새로운 상태에서 최적의 행동을 찾아 Q값을 업데이트 한다. 최선의 선택만 고집하여 함정에 빠지지 않도록 $\epsilon-greedy$ 전략을 추가한 것이 Q-learning이다(Knox *et al.*, 2010; Fan *et al.*, 2019).

2.2 Deep Q Networks

Deep Q Networks 알고리즘은 Q-learning을 심층 신경망으로 학습한 것이다. Hasselt *et al.*(2016) DQN을 게임에 적용할 경우에 Convolution Neural Network(CNN)을 통해서 중요한 정보를 자동으로 출력하고, 그 feature들을 기반으로 다시 각각의 Q값을 계산 하기때문에 feature도 자동으로 뽑아내며 작은 상태 변화에 대해 로버스트한 계산을 기대할 수 있다(Hasselt, 2016). 대표 구조는 <Figure 1>과 같다.

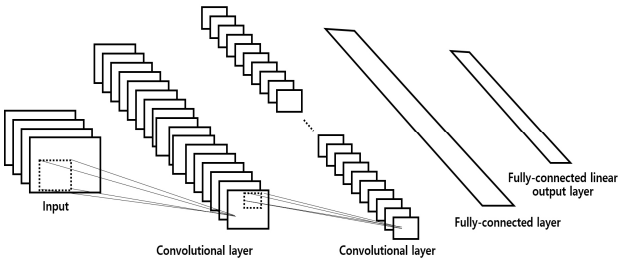

Figure 1. Structure of DQN with CNN

입력 데이터에 여러 겹의 파라미터를 붙혀서 최종적으로 결과 값을 도출하는 DNN을 DQN에 적용할 수 있다. 강화학습 과정에서 입력데이터에 대한 feature들을 추출하고, 그 feature들을 다시 한 번 Fully connected layer를 통해서 결과 값을 도출하는 방식이다. 모든 데이터를 함께 고려하여 계산을 한다는 점에서 시계열 데이터에서 적합한 결과를 기대할 수 있다. 대표 구조는 <Figure 2>와 같다.

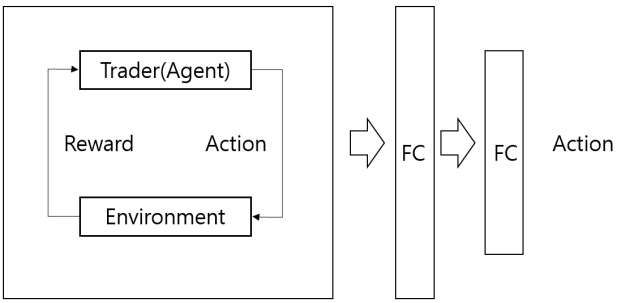

Figure 2. Structure of DQN with DNN

2.3 Policy Gradient

정책경사는 수식에서 보통 π라고 표현되는 정책를 직접적으로 모델링하고 최적화하는데 주력하며, 보통은 θ라고 하는 특정 parameter로 구성된 함수, π_θ로 표현 된다. reward(objective) function의 값은 이 정책에 영향을 받고, 해당 알고리즘을 적용한 방법들은 보통 최적의 보상을 얻는데 있어서 이 θ를 최적화하는 경향을 띄고 있다. 보상함수는 식 (3)과 같이 정의 된다.

$$\nabla_\theta J(\pi_\theta) = \sum_{s\in S} d^\pi(s)\, V^\pi(s) = \sum_{s\in S} d^\pi(s) \sum_{a\in A} \pi_\theta(a \mid s)\, Q^\pi(s,a) \quad (3)$$

알고리즘은 θ에 대해 목적함수를 미분한 $\nabla_\theta J(\pi_\theta)$ 방향으로 파라미터 θ를 조절한다.

$$\nabla_\theta J(\pi_\theta) = \int_s \rho^\pi(s) \int_A \nabla_\theta \pi_\theta(a|s)\, Q^\pi(s,a)\, da\, ds = E_{s\sim\rho^\pi, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)\, Q^\pi(s,a)\right] \quad (4)$$

여기서 $Q^\pi(s,a)$는 장기간의 보상을 고려한 상태 s에서 행동 a를 수행했을 때의 가치다. 정책 경사는 다양한 방식으로 손실함수를 정할 수 있다. 다음에 이어서 살펴볼 Actor-Critic, A2C 등이 정책 경사 방식의 변형이라고 볼 수 있다(Silver *et al.*, 2014).

2.4 Advantage Actor-Critic

Actor-Critic 모델은 Actor 네트워크와 Critic 네트워크 두 개의 네트워크를 사용한다. Actor는 상태가 주어졌을 때 행동을 결정하고, Critic은 상태의 가치를 평가한다(Barto *et al.*, 1983). 앞서 나타난 에이전트의 행동확률을 직접적으로 학습하는 정책 경사 방식을 변형하여, 에이전트의 행동확률을 직접적으로 학습하는 것이 불안정하다는 가정하에 가치함수를 같이 써서 안정성을 높이는 것이 Actor-Critic 모델이다.

Actor-Critic 모델에서 기대 출력을 Advantage로 사용하는 A2C(advantage actor-critic)이다. Actor-Critic은 Actor와 Critic 두 가지 모델로 구성되어 있다. 알고리즘에서 사용하는 Action-value-function을 $Q_\omega(a|s)$라 할 때, Critic은 파라미터 ω를 업데이트한다. 그리고 현재의 Policy를 $\pi_\theta(a|s)$라 할 때, Actor는 Critic이 결정하는 방향으로 파라미터 θ를 업데이트한다. 즉, Critic은 Q function을 추정하고 Actor는 Policy를 추정한다(Konda *et al.*, 2000). Advantage는 상태-행동 가치에서 상태 가치를 뺀 값이며, 식으로 표현하면 식 (5)와 같다.

$$D(s,a) = Q(s,a) - V(s) \quad (5)$$

Advantage를 적용한 손실함수는 (6)과 같이 바뀌게 된다.

$$L(\theta) = -E\left[\nabla_\theta \log \pi_\theta(a|s)\, Q_e(s,a)\right] \quad (6)$$

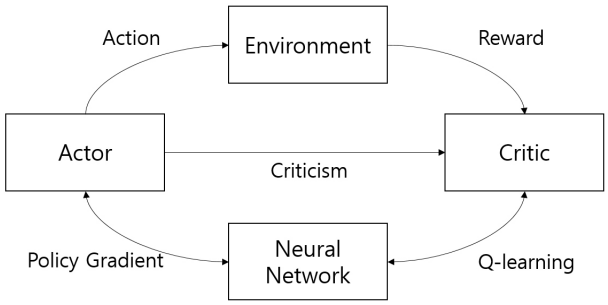

Figure 3. Framework of A2C

A2C에서는 Critic을 Advantage로 학습한다.

A2C(advantage actor-critic)를 사용하면 어떤 상태에서 행동이 얼마나 좋은지 뿐만 아니라 얼마나 더 좋아지는지를 학습할 수 있게 된다(Xiong *et al.*, 2018; Li, 2018).

3. 모 델

본 논문에서 설계한 강화학습 모델의 구성은 <Table 1>과 같다. 실험에 사용한 강화학습 기법은 3가지를 사용했다. DQN, PG, A2C를 사용하였는데. 주가 예측과 같이 변동성의 큰 데이터의 경우에는 경험을 많이 쌓아 놓아도 정책이 업데이트되면 쌓아놓은 경험들은 학습에 사용할 수 없는 on-policy 보다 행동하는 정책과 학습하는 정책이 나눠서 진행되는 off-policy 기법을 이용하기 위해서 모델을 선정하였고, 그 중에 많은 종류의 신경망 모델을 사용할 수 있는 3가지 기법을 선정하여 실험을 진행했다. 모델을 학습하는 과정은 <Figure 2>와 같이 DNN을 이용한 강화학습 모델을 적용했다. 사용한 DNN의 구성은 <Figure 4>와 같다.

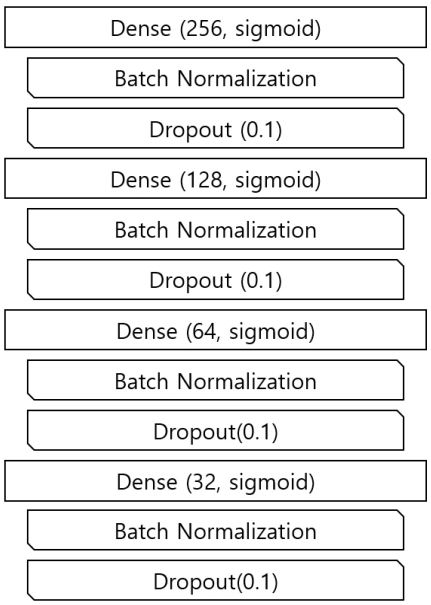

Figure 4. DNN Architecture for Reinforce Learning Model

주어진 상태(state)는 시가, 고가, 저가, 종가, 거래량을 기반으로 거래량의 이동평균비율, 종가의 이동평균비율을 추가로 산출한 데이터를 기초로 한다. 일별로 상태(state) 데이터에 따라 주체(agent)는 강화학습 기법을 바탕으로 행동(action)을 결정한다. 학습 초기에는 탐험률(ϵ)에 따라 매수/매도/관망의 무작위 행동을 취한다. 신경망 학습이 진행될수록 무작위 행동은 줄어들고 학습된 신경망에 의해 결정된 행동을 취한다. 환경(environment) 안에서 행동의 결과로 PV(Portfolio Value)가 정해지는데, PV의 변동이 임계점을 초과했을 때 임계점을 초과하기까지의 행동들로 하나의 batch를 생성하고 보상을 부여한다. 예를 들어 임계점이 0.1인 경우 10번의 행동을 수행한 후에 PV가 10% 상승하거나 하락할 때 10번의 행동에 대한 보상을 부여한다. 이때 발생한 보상을 바탕으로 신경망을 업데이트한다.

행동을 결정하는 정책 신경망과 주체의 행동을 평가하는 가치 신경망 모두 DNN을 기반으로 하였다. DNN의 optimizer는 확률적 경사하강법(stochastic gradient descent)로 하였고, 과적합을 막기 위해 Dropout(0.2)을 활용하였다. 손익률을 선형회귀모형을 가치 신경망으로 활용하는 모델이기 때문에 activation은 선형 함수로, 손실 함수는 MSE로 설정하였고 정책 신경망의 경우 PV를 높이기 위해 취하기 좋은 행동에 대한 분류 모델이기 때문에 활성화 함수 activation 인자로 'sigmoid'를 사용하였다.

강화학습 기법에 대한 조건과 모델 파라미터는 <Table 1>과 같다. 초기 탐험비율은 1로 설정해서 학습되지 않은 경우에 탐험하는 비율을 크게하여 행동을 결정하기위한 경험을 쌓도록 하였다. 이 값은 학습이 진행되면서 감소하게 된다. 주식투자 시뮬레이션을 진행하기 위해서 설정한 초기 자본금은 10,000,000으로 설정하였고, 할인율은 0.9로 설정하였다. 할인율은 상태-행동 가치를 구할 때 적용할 할인율이다. 지연 보상 임계값은 0.04로 설정했고, 손익률이 이 값을 넘으면 지연 보상이 발생하도록 하였다.

Table 1. Model Configuration and Description

	Options
Environment	Stock price data
State	Point of Stock price data, Close price moving average ratio
Agent	Buying/Selling/Hold
Reword	Revenue Beyond Critical Points Loss occurrence
Reinforcement learning method	Deep Q Networks, Policy Gradient, Advantage actor-critic
Neural network	DNN
Learning rate	0.01
Initial exploration rate	1
Initial capital	10,000,000
num of epochs	3
Discount factor	0.9
Delayed reward threshold	0.04

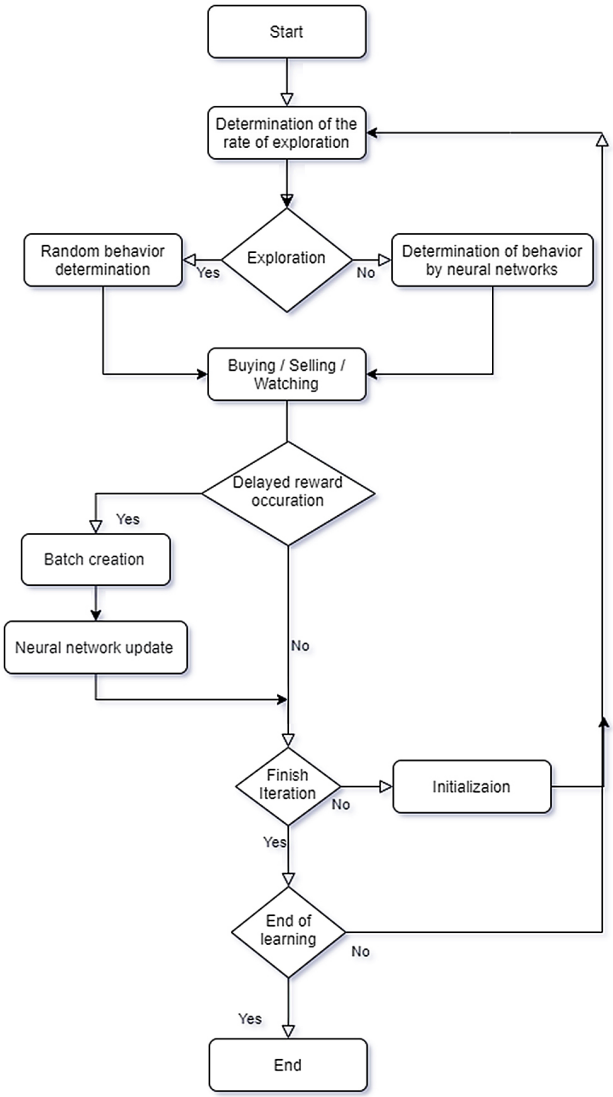

Figure 5. Model Framework

4. 실험 결과

모델의 성능을 평가하기 위해 수익의 평균을 수익성의 평가지표로, 손실이 발생한 종목의 손실 평균을 안정성 평가지표로 설정하였다. 수익률은 초기 투자한 원금 대비해서 수익을 낸 비율이고, 손실평균은 원금 대비 손실이 일어난 종목들의 평균값이다. 모델에 적용할 종목으로 시가총액 상위 20개 종목, 바이오 테마주 20개 종목, 언택트 테마주 20개 종목을 선정하여 실험 하였다. 종목의 구성은 <Table 2>와 같다.

총 3가지 비교 실험을 진행하였다. 첫 번째로는 DQN과 PG, A2C 3가지 강화학습 기법의 성능을 비교하는 실험을 진행했다. 두 번째 실험은 학습기간에서의 변동성에 따른 실험을 진행하였고, 세 번째 실험은 학습기간의 차이에 대한 실험결과를 비교하였다. 첫 번째 실험에서 가장 높은 수익률을 나타낸 모델을 이용하여, 두 번째, 세 번째 실험을 진행하였다.

Table 2. Used items for Datasets

	KOSPI	BIO	Untact
1	KB금융	SK바이오랜드	다날
2	LG생활건강	네이처셀	SGA솔루션즈
3	LG전자	동국제약	KG모빌리언스
4	LG화학	랩지노믹스	아이씨케이
5	NAVER	레고켐바이오	이니텍
6	POSCO	비씨월드제약	시큐브
7	SK	삼천당제약	카페24
8	SK텔레콤	신풍제약	NHN한국사이버결제
9	SK하이닉스	수젠텍	인포뱅크
10	기아차	CMG제약	인포바인
11	넷마블	안트로젠	케이씨에스
12	삼성SDI	일신바이오	씨아이테크
13	삼성물산	일양약품	한국전자금융
14	삼성바이오로직스	진원생명과학	윈스
15	삼성전자	차바이오텍	에이택티앤
16	셀트리온	코미팜	파인디지털
17	엔씨소프트	텔콘RF제약	푸른기술
18	카카오	파미셀	디지탈옵틱
19	현대모비스	피씨엘	KG이니시스
20	현대자동차	현대바이오	아이크래프트

4.1 DQN, PG, A2C 성능을 비교

모델의 학습 기간은 2017. 09. 01~2019. 08. 31, 2년으로 테스트 기간은 2019. 09. 01~2020.08. 31, 1년으로 하였다.

실험의 결과는 <Table 3>과 같다. 가장 수익성이 높은 강화학습 기법은 A2C으로 시가총액 상위 20종목은 9.66%, 바이오 테마주는 46.97%, 언택트 테마주는 17.52%이다. DQN은 수익성은 떨어지지만, 안정성이 높은 학습 기법으로 손실 종목의 평균이 1.63%, 0.4%, 0.91%로 다른 기법에 비해서 손실이 매우 낮다.

Table 3. Model Performance by Reinforcement Learning Techniques

	Models	Profits	Loss average
KOSPI	A2C	9.66%	-5.88%
	PG	7.64%	-5.50%
	DQN	4.24%	-1.63%
BIO	A2C	46.97%	-7.44%
	PG	1.81%	-0.01654%
	DQN	3.54%	-0.40%
Untact	A2C	17.52%	-10.20%
	PG	15.57%	0.00%
	DQN	5.28%	-0.91%

4.2 학습 기간의 변동성에 따른 모델 안정성 비교

코스피 변동성 지수(KOSPI Volatility Index)를 기준으로 VIX 지수가 20을 초과하는 기간을 변동성이 높은 기간으로 VIX 지수가 20 이하인 기간을 변동성이 낮은 기간으로 설정하였다. 실험은 2020. 01. 01. 이전의 데이터를 낮은 변동성(Volarilty-low) 기간으로, 2020. 01. 01. 이후의 데이터를 높은 변동성(Volarity-high) 기간으로 설정하였고, <Table 4>와 같이 두 데이터를 교차하여 총 4가지 방법으로 진행하였다. 적용한 강화학습 기법은 위 비교실험에서 가장 수익성이 높았던 기법인 A2C을 적용하였다.

실험의 결과는 <Table 4>와 같다. 변동성이 높은 시기에 학습한 모델은 VIX-low test에서 평균 -4.66% VIX-high test에서 평균 -4.59%이고 변동성이 낮은 시기에 학습한 모델은 VIX-low test에서 평균 -13.09% VIX-high test에서 평균 -7.84%로 변동성이 높은 구간에서 학습한 모델이 더 안정성이 높다는 결과를 얻었다.

Table 4. Train Set and Test Set According to Variability

		VIX-low test		VIX-high test	
Loss average		Profits	Loss average	Profits	Loss average
VIX-low train	KOSPI	2.05%	-9.71%	9.66%	-5.88%
	BIO	-1.11%	-18.09%	46.97%	-7.44%
	Untact	-2.04%	-11.49%	17.52%	-10.20%
	Average	-0.37%	**-13.09%**	24.72%	**-7.84%**
VIX-high train	KOSPI	2.28%	-2.98%	5.93%	-0.31%
	BIO	7.50%	-6.95%	25.01%	-11.40%
	Untact	0.87%	-4.04%	8.03%	-2.06%
	Average	3.55%	**-4.66%**	12.99%	**-4.59%**

진행한 4번의 실험 결과의 평균을 계산하여 시가 총액 상위 20종목에서 안정성이 가장 높은 종목을 확인해 보았다. 코스피가 가장 안정성이 높았고, 전체적인 결과는 <Figure 6>과 같다.

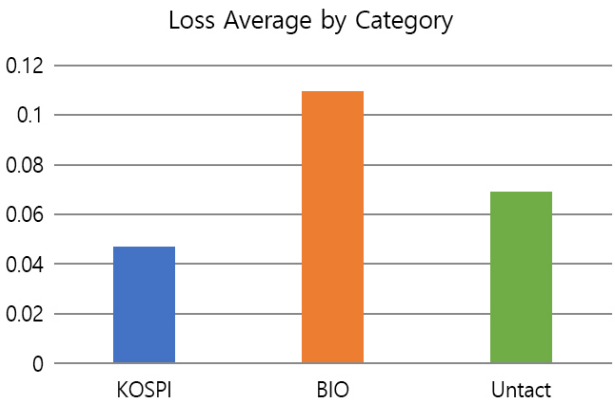

Figure 6. Average Loss by Category

4.3 모델의 학습 기간 길이별 성능 비교

10년 학습 모델과 2년 학습 모델의 성능을 비교하였다. 두 실험의 데이터 분리는 <Figure 7>과 같이 설정하였다. 10년 학습 모델의 학습 기간은 2009. 01. 01.~2019. 08. 31, 2년 학습 모델의 학습 기간은 2017. 09. 01~2019. 08. 31이다. 테스트 기간은 2019. 09. 01~2020. 08. 31, 1년으로 동일하다. 10년간 주가 데이터가 존재해야 하므로 시가총액 상위 20종목 중에서 데이터가 존재하는 19개 종목으로 실험하였다. 적용한 강화학습 기법은 1번 비교실험에서 가장 수익성이 높았던 기법인 A2C을 적용하였다.

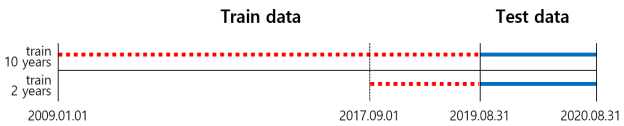

Figure 7. Data Distribution of Learning Period Performance Comparison Experiments

실험의 결과는 <Table 5>와 같다. 10년 데이터를 학습한 모델은 27.79%의 수익을 냈고 2년 데이터를 학습한 모델은 11.2%의 수익 냈다. 10년 데이터를 학습한 모델의 손실 평균은 -6.02%이고 2년 데이터를 학습한 모델의 손실 평균은 -2.25%로 2년 데이터를 학습한 모델의 안정성이 더 높다.

Tabel 5. Comparison of performance by length of learning period

	Avg of profits	Avg of lost stocks
Training 10 years	27.79%	-6.02%
Training 2 years	12.28%	-2.25%

5. 결 론

머신러닝을 통한 예측 알고리즘이 대부분의 도메인에서 많이 사용되고 있다. 그런데 주가 예측의 같은 경우에 대부분의 연구들이 특정 일자를 기준으로 예측을 진행하기 때문에 실시간으로 즉각적인 의사 결정이 요구되는 자동매매 시스템이 직접적으로 응용하는 데에는 한계를 가지고 있다. 따라서 즉각적으로 이루어지는 매도/매수 의사결정을 고려했을 때, 보상과 행동을 모델링할 수 있는 강화학습이 효과적인 대안이 될 수 있다.

본 논문에서는 세 가지 비교 실험을 통해 강화학습 기반 주식 자동매매 시스템 사용자의 목적에 맞는 강화학습 전략을 제안한다. 첫 번째로 DQN과 PG, A2C 세 가지 강화학습 알고리즘을 같은데이터를 이용하여 비교 실험을 진행했다. 결과는 시스템의 사용자가 수익성과 안정성 중에 무엇을 더 중요시하는가에 따라서 모델을 달리 설계할 수 있다. 사용자가 수익성이 중요하다면, A2C 학습 기법을 기반으로 하고 학습 기간을 길게 설정하는 것이 더 높은 수익성을 기대할 수 있다. 안정성을 중요시한다면 DQN과 PG 학습 기법을 기반으로 학습 기간을 변동성이 높은 구간 위주로 학습시키며 학습 기간을 짧게 설정하는 것이 더 높은 안정성을 기대할 수 있다.

두 번째로 훈련 데이터와 테스트 데이터를 변동성이 큰 데이터와 작은 데이터의 조합으로 나누어서 비교 실험을 진행했다. 변동성이 높은 주가 데이터의 특성을 고려해봤을 때 변동성 높은 데이터로 모델을 학습했을 경우가 좀 더 안정성이 높다는 결론을 지었다.

세 번째로 학습 기간 길이에 차이를 두고 진행한 실험에서는 학습기간을 결정함에 있어 학습 기간 길이가 길수록 수익과 위험이 모두 높게 측정되는 이유는 오랜 기간을 학습할수록 같은 패턴에 대해서 더 많은 빈도로 학습되기 때문에 신뢰도를 높게 판단하여 더 많은 단위로 주식을 매매하기 때문이라고 판단된다. 따라서 해당 시스템에 한하여 학습 기간을 길게 가져갈수록 수익성은 커지고 안정성은 낮아진다고 결론 지었다.

본 연구의 좁은 범위에서의 실험으로 범용성이 부족하다는 한계를 갖는다. 또한 모델이 종가로 매수한다는 가정하에 학습과 테스트를 하므로 실제 매매와의 간극이 존재한다. 추후 연구로 해당 모델의 입력 데이터로 기본적인 주가 데이터 이외에 주가 예측에 유효한 기술적 분석 지표 등을 추가하여 성능을 개선하거나 가장 효율적인 파라미터를 탐색하여 성능을 개선할 수 있다. 또 추세별 학습 모델의 성능을 비교하거나 주식 외에도 증권 시장에 연동이 되어있는 다양한 상품에도 적용해 자산 포트폴리오를 다양하게 구성할 수 있을 것으로 보인다.

참고문헌

Adebiyi, A. A., Adewumi, A. O., and Ayo, C. K. (2014), Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction, *Journal of Applied Mathematics*, **2014**, 1-7.

Adebiyi, A. A. (2014), Stock Price Prediction Using the ARIMA Model, *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation,* IEEE.

Azhikodan, A. R., Baht, A. G. K., and Jadhav, M. V. (2019), Stock Trading Bot Using Deep Reinforcement Learning, *Innovations in Computer Science and Engineering*, 41-49.

Barto, A. and Sutton, R. (1983), Neuron-Like Elements that can Solve Difficult Learning Control Problems, *IEEE Transactions on Systems, Man, and Cybernetics*, 834-846.

Chen, S. and He, H. (2017), Stock Prediction Using Convolutional Neural Network, *Artificial Intelligence Applications and Technologies* (AIAAT 2018).

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017), Deep Direct Reinforcement Learning for Financial Signal Representation and Trading, *IEEE Transactions on Neural Networks and Learning Systems* **28**(3), 653-664.

Fan, J., Wang, Z., Xie, Y., and Yang, Z. (2019), A Theoretical Analysis of Deep Q-Learning, Proceedings of Machine Learning Research,

*arXiv preprint arXiv:*1901.00137.
Hasselt, H., Guez, A., and Silver, D. (2016), Deep Reinforcement Learning with Double Q-learning, *Thirtieth AAAI Conference on Artificial Intelligence*, **30**(1).
Knox, W. and Stone, P. (2010), Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning, *9th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS.
Konda, V. R. and Tsitsiklis, J. N. (2000), Actor-Critic Algorithms, *In Advances in Neural Information Proceeding Systems, Neural Information Processing Systems*, 1008-1014.
Lee, J.-H. (2017), Stock Price Prediction Model Using Deep Learning, Soongsil University.
Lee, J.-W. (2001), Stock Price Prediction Using Reinforcement Learning, *2001 IEEE International Symposium on Industrial Electronics Proceedings* (Cat. No.01TH8570).
Lee, J.-W., Kim, S.-D., Lee, J.-W., and Chae, J.-S. (2002), R-Trader : An Automatic Stock Trading System based on Reinforcement Learning, *Journal of KISS : Software and Applications*, **29**(1112), 785-794.
Li, S., Bing, S., and Yang, S. (2018), Distributional Advantage Actor-Critic, *arXiv preprint arXiv:*1901.00137.
Mckenzie, M., Loxley, P., Billingsley, W., and Wong, S. (2017), Competitive Reinforcement Learning in Atari Games, *AI 2017 : Advances in Artificial Intelligence*, 14-26.
Shen, S. and Jiang, H. (2012), Stock Market Forecasting Using Machine Learning Algorithms, Citeseer.
Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014), Deterministic Policy Gradient Algorithms, *International Conference on Machine Learning*, **32**(1), 387-395.
Singh, R. and Srivastava, S. (2017), Stock Prediction Using Deep Learning, *Multimedia Tools and Applications*, **76**, 18569-18584.
Won, J.-M., Hwang, H.-S., Jung, Y.-H., and Park, H.-D. (2018), Stock Price Prediction Technique Using Technical Analysis Index and Deep learning, *Proceedings of KIIT Conference*, 404-405.
Xiong, Z., Liu, X.-Y., Zhong, S., Yang, H., and Walid, A. (2018), Practical Deep Reinforcement Learning Approach for Stock Trading, Neural Information Processing Systems, *arXiv preprint arXiv:*1811.07522.

저자소개

황호현 : 서울과학기술대학교 산업공학과에서 2020년 학사학위를 취득하고 일반대학원 데이터사이언스학과 석사과정에 재학중이다. 연구분야는 데이터 마이닝, 인공 신경망이다.

김용훈 : 서울과학기술대학교 산업공학과에서 2021년 학사학위를 취득했다. 연구분야는 강화학습, 금융데이터 분석이다.

이영훈 : 서울대학교 산업공학과에서 2007년 학사, 2009년 석사, 2019년 박사학위를 취득하였다. 현대자동차 빅데이터실 책임연구원, LG전자 UX연구소/선행디자인연구소 선임연구원을 역임하고, 2019년부터 서울과학기술대학교 산업공학과 교수로 재직하고 있다. 연구분야는 데이터마이닝, 사용자경험(UX) 디자인이다.