Pusan National University

Computer Science and Engineering

Technical Report 2023-10

심층강화학습을 이용한 주식투자 전략 개발

저자 1 : 정희영

저자 2 : 신재환

저자 3 : 박동진

지도교수 : 유영환

목 차

1. 서론

1.1. 연구 배경

2022년 11월 처음으로 세상에 공개된 ChatGPT는 미국의 AI 회사인 Open AI에서 개발한 인공지능 서비스로 최근 전세계적으로 이슈가 되고 있다. ChatGPT 이전에도 많은 대화형 인공지능 서비스가 존재해왔지만, ChatGPT는 다른 인공지능과 달리 인터넷에서 답을 찾지 않고 지도학습(Supervised Learning)과 강화학습(Reinforcement learning)을 통해 데이터를 학습한다. 강화학습은 결과를 극대화하기 위해 시행착오 방법을 사용하여 동적으로 학습하는데, 이를 활용해 주식투자에 적용하면 실용적일 것이라 판단되었다. 따라서, 기존 지식에서 학습하여 새로운 데이터셋에 적용할 수 있는 심층 강화 학습(Deep Reinforcement Learning, DRL)을 사용하여 주식투자 전략 개발을 해보고자 한다.

1.2. 기존 문제점

기존에 제공되던 인공지능 기반 주식추천 서비스들은 유료로 제공되거나, 투자자가 대행업체에게 투자금을 맡겨 투자자가 대행업체에게 투자금을 위탁하여 운용되는 형태다. 이러한 방식은 투자자가 자신의 돈을 제3자에게 위탁하고, 과거의 투자 기록을 투명하게 확인할 수 없는 문제점을 갖고 있다.

본 과제는 위의 문제점을 해결하기 위해 이용자에게 강화학습 모델의 결과와 과거 예측 내용을 함께 무료로 제공하여, 이를 토대로 투자자가 직접 결정할 수 있도록 한다.

1.3. 연구 목표

지금까지 개발된 다양한 강화학습 기법과 신경망들을 이용해 강화학습을 수행하여, 주식투자 전략에 가장 적합한 강화학습 기법을 발견한다. 채택된 기법에 대해서는 다양한 hyperparameter set과 데이터셋을 이용하며 모델을 생성, 평가하여 가장 적합한 매개변수를 판별해 가장 적합한 모델을 개발한다. 강화학습 모델의 결과를 웹으로 제공함으로써 투자자에게 가치 있는 의사 결정 도구를 제공하는데 기여하고자 한다.

2. 연구 배경

강화학습은 인공지능 분야에서 높은 성능과 다양한 응용 가능성을 보이고 있지만, 실제로 강화학습 모델을 개발하고 효과적으로 학습시키는 것은 매우 어려운 과제이다. 강화학습 모델을 구축하는 과정에서는 다양한 환경 설정, 알고리즘 선택, 데이터 처리, 하이퍼파라미터 조정 등 다양한 변수를 고려해야 한다. 이러한 다양한 요인들을 고려하면서

강화학습 모델을 개발하려면 많은 시간과 노력이 요구된다.

OpenAI에서 제공하는 라이브러리인 OpenAI Gym을 이용하여 필요한 강화학습 환경을 어렵지 않게 개발할 수 있다. 또한 Stable-Baselines3를 이용하면 Open AI Gym을 이용해 개발한 환경에서 다양한 알고리즘을 이용하여 에이전트를 학습시킬 수 있다. 이러한 도구들을 이용하여 많은 시행착오를 줄이고 강화학습 모델의 개발과 실험을 더욱 효율적으로 진행할 수 있게 되었다.

2.1. 개발 방법 및 도구

데이터 수집

국내 증권사의 API, 웹 크롤링, 제공된 라이브러리를 이용하여 주가에 영향을 미칠 만한 데이터를 수집한다.

환경 모델링

Open AI Gym을 이용하여 주식 시장 환경을 모델링한다. 관측(Observation Space)과 행동(Action Space), 보상 함수(Reward)를 정의하고, 강화학습 에이전트가 투자 결정을 내릴 수 있는 환경을 구성한다. 다양한 환경으로 시행착오를 거쳐 주가예측에 가장 효과적인 환경을 발견한다.

에이전트 학습을 통한 모델 개발

Stable-Baselines3 라이브러리를 이용하여 주식투자를 위한 강화학습 모델을 구현한다. 다양한 알고리즘을 실험하고 성능을 평가하여 최적의 모델을 선택한다.

결과 제공을 위한 웹 개발

매일 업데이트 되는 주식시장의 정보에 따라 모델의 예측을 확인할 수 있도록 웹 어플리케이션을 개발하여 사용자에게 정보를 제공한다. 웹사이트 구성을 위해서는 React프레임워크를 API응답을 위해서는 AWS로 배포한 Spring Boot를 이용한다.

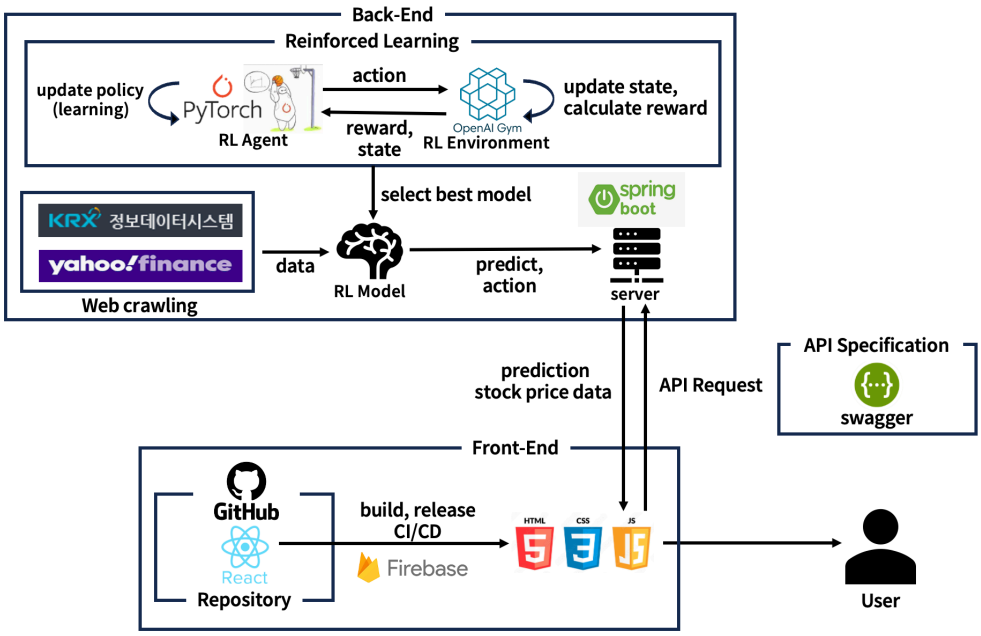


Figure 1. 프로젝트 계통도

3. 연구 내용

3.1. 학습 데이터 수집

심층 강화학습을 이용한 주식투자전략 개발이라는 주제에 맞추어 강화학습을 사용하였을 때, 효율적으로 투자전략을 개발할 수 있는 주식 관련 데이터들을 선정하여, 웹크롤링을 진행하였다.

3.1.1. 사용한 데이터

A. 기본적 분석 지표

주가수익비율(Price Earning Ratio, PER)

PER은 주가와 회사의 순이익의 비율이다. PER이 낮을수록 회사의 순이익에 비해 주가가 저평가돼 있다고 볼 수 있다. 그러나 업종마다 평균 PER의 차이가 크기 때문에 PER이 높다, 낮다라고 절대적으로 판단할 기준은 없다.

주가순자산비율(Price Book-value Ratio, PBR)

PBR은 자산 대비 주가의 비율이다. PBR이 낮을수록 회사의 가치 대비 주가가 낮다는 것이므로 해당 종목이 저평가돼 있을 가능성이 있다.

주당순자산가치(Bookvalue Per Share, BPS)

BPS는 기업이 활동을 중단한 뒤 그 자산을 모든 주주들에게 나눠줄 경우 1주당 얼마씩 배분되는가를 나타낸다. 기업의 총자산에서 부채를 빼면 기업의 순자산이 남는데, 이 순자산을 발행주식수로 나눈 수치이다. BPS가 높을수록 수익성 및 재무건전성이 높아 투자가치가 높은 기업이라 생각할 수 있지만 BPS 지표를 사용하기보다는 주가와 청산 가치 사이의 비율을 알 수 있는 PBR 값을 이용하는 것이 더 효과적이다.

B. 기술적 분석 지표

당일 종가 대비 전일 종가 비율(close/last close)

당일 종가를 전일 종가로 나눈 값이다. 일반적으로 주가가 올랐다, 떨어졌다를 얘기할 때 당일 종가에서 전일 종가를 뺀 다음에 다시 전일 종가로 나누는데, 수식으로는 (close-last close/last close)가 된다.

C. 기본 제공 데이터

환율(exchange rate)

'한 나라의 화폐와 외국 화폐의 교환 비율로 외화 1단위와 교환되는 원화의 양을 의미한다. 달러 환율의 경우 주가에 큰 영향을 미치기에 학습 데이터로 선정하였다.

코스피, 나스닥 지수

개발한 주식투자전략의 정확도 및 수익률을 테스트하기 위한 주식으로 삼성전자, SK하이닉스, POSCO홀딩스를 선택하였는데, 이 세 종목의 경우 코스피 200에서 아주 높은 비중을 차지하고 있다. 따라서, 이 종목 중 한가지 종목이 상승/하락할 때 코스피 200이 상승/하락하기도 하지만 코스피 200의 상승/하락에 따라서 세가지 종목 모두가 상승/하락하는 경우도 많기에 학습시키고자 하였다. 미국 연방준비제도(연준) 의장인 제롬 파월의 연설에 따라 나스닥 지수가 크게 영향을 받게 되는데, 국내 주가의 경우에도 나스닥 지수의 큰 영향을 받기에 수집하였다.

주당배당금, 배당수익

주당배당금은 보유 주식 한 주당 지급되는 배당금액으로 배당금 = 보유주식수 x 주당배당금이다. 배당수익률은 현 주가에 비해 배당금이 얼마나 많은지를 확인할 수 있는 지표로 배당수익률 = (주당배당금 / 주가) x 100%이다. 배당락(배당 기준일 직후 주가가 떨어지는 현상)이 있을 정도로 배당주 투자를 하는 주주들도 많기 때문에 주당배당금과 배당수익률 또한 주가에 영향을 미칠 것으로 판단하였다.

3.1.2. 데이터 수집 방법

데이터를 수집하기 위하여 필요한 데이터인 Date, Close(종가), 대비(전날 대비 주가 변동 가격), BPS, PBR, 주당배당금, 배당수익률을 제공하는 KRX 정보데이터시스템과 Date, Close(종가), PER, 환율, 코스피 지수, 나스닥 지수를 제공하는 야후 파이낸스에서 웹 크롤링을 해오는 프로그램을 개발하였다. 키움증권 API를 사용하려고 하였으나 키움증권 API의 경우에는 날짜에 따른 BPS, PER, PBR 데이터를 받아올 수가 없었고 KRX 정보데이터 시스템은 주식 정보를 활용한 프로그램 개발을 위해 만들어진 시스템이 아니다보니 서버 자체에서 IP를 차단하는 문제가 있었다. 이 부분은 반기 단위로 데이터를 수집하도록 하여 문제를 해결하였다.

```
Date,Close,대비,등락률,BPS,PBR,주당배당금,배당수익률
2014/01/02,322000,-4500,-1.38,452524,0.71,8000,2.48
2014/01/03,320500,-1500,-0.47,452524,0.71,8000,2.50
2014/01/06,316500,-4000,-1.25,452524,0.70,8000,2.53
2014/01/07,314500,-2000,-0.63,452524,0.69,8000,2.54
2014/01/08,313500,-1000,-0.32,452524,0.69,8000,2.55
2014/01/09,306500,-7000,-2.23,452524,0.68,8000,2.61
2014/01/10,308000,1500,+0.49,452524,0.68,8000,2.60
2014/01/13,310500,2500,+0.81,452524,0.69,8000,2.58
2014/01/14,310000,-500,-0.16,452524,0.69,8000,2.58
2014/01/15,310000,0,0.00,452524,0.69,8000,2.58
2014/01/16,311500,1500,+0.48,452524,0.69,8000,2.57
2014/01/17,311500,0,0.00,452524,0.69,8000,2.57
2014/01/20,311000,-500,-0.16,452524,0.69,8000,2.57
2014/01/21,309500,-1500,-0.48,452524,0.68,8000,2.58
2014/01/22,310500,1000,+0.32,452524,0.69,8000,2.58
2014/01/23,305000,-5500,-1.77,452524,0.67,8000,2.62
2014/01/24,304500,-500,-0.16,452524,0.67,8000,2.63
2014/01/27,299000,-5500,-1.81,452524,0.66,8000,2.68
2014/01/28,297500,-1500,-0.50,452524,0.66,8000,2.69
2014/01/29,298500,1000,+0.34,452524,0.66,8000,2.68
```

Figure 2. KRX 정보데이터 시스템 크롤링 데이터

```
Date,Close,등락률,PER,PBR,회전율,환율,코스피,나스닥
2014-01-02,322000,-1.38,10.1,0.71,0.002,1050.75,1967.19,-0.8
2014-01-03,320500,-0.47,10.06,0.71,0.002,1049.5999755859375,1946.14,-0.27
2014-01-06,316500,-1.25,9.93,0.7,0.002,1053.800048828125,1953.28,-0.44
2014-01-07,314500,-0.63,9.87,0.69,0.0016,1065.4000244140625,1959.44,0.96
2014-01-08,313500,-0.32,9.84,0.69,0.0018,1068.5,1958.96,0.3
2014-01-09,306500,-2.23,9.62,0.68,0.0034,1064.699951171875,1946.11,-0.23
2014-01-10,308000,0.49,9.66,0.68,0.0018,1062.0,1938.54,0.44
2014-01-13,310500,0.81,9.74,0.69,0.0017,1060.4000244140625,1948.92,-1.47
2014-01-14,310000,-0.16,9.73,0.69,0.0014,1058.5,1946.07,1.69
2014-01-15,310000,0.0,9.73,0.69,0.0014,1058.300048828125,1953.28,0.76
2014-01-16,311500,0.48,9.77,0.69,0.002,1063.0,1957.32,0.09
2014-01-17,311500,0.0,9.77,0.69,0.0017,1062.4000244140625,1944.48,-0.5
2014-01-21,309500,-0.48,9.71,0.68,0.0016,1062.5,1963.89,0.67
2014-01-22,310500,0.32,9.74,0.69,0.0016,1064.199951171875,1970.42,0.41
2014-01-23,305000,-1.77,9.57,0.67,0.0036,1066.5,1947.59,-0.57
2014-01-24,304500,-0.16,9.55,0.67,0.0022,1072.699951171875,1940.56,-2.15
2014-01-27,299000,-1.81,9.38,0.66,0.0031,1078.4000244140625,1910.34,-1.08
2014-01-28,297500,-0.5,9.33,0.66,0.0021,1082.5,1916.93,0.35
2014-01-29,298500,0.34,9.36,0.66,0.0032,1076.9000244140625,1941.15,-1.14
2014-02-03,295000,-1.17,9.26,0.65,0.003,1080.199951171875,1919.96,-2.61
```

Figure 3. Yahoo Finance 크롤링 데이터

3.2. 강화학습 모델

강화 학습(reinforcement learning)은 기계 학습의 한 분야로. 특정 환경 안에서 정의된 에이전트가 현재의 상태를 인식하여, 선택 가능한 행동들 중 보상을 최대화하는 행동 또는 행동 순서를 선택하는 방법이다. 강화학습을 통해 유효한 예측모델을 생성하기 위해서는 적절한 환경과 똑똑한 에이전트가 필요하다.

3.2.1. 강화학습 에이전트

강화학습 에이전트는 주어진 환경과 상호작용하면서 행동을 결정하는 의사 결정 주체라고 볼 수 있다. 에이전트는 현재 자신의 상태 및 주변 상태를 파악하고 정책(policy)을 통해 자신이 취할 최적의 행동을 선택한다. 강화학습에서 정책이란 에이전트의 행동 패턴,

즉 어떤 state에 대해 어떤 action을 선택할지를 정의하는 것이다.

강화학습 알고리즘에는 여러 종류가 있지만 본 과제에서는 DQN(Deep Q-Network) 알고리즘과 A2C(Advantage Actor-Critic) 알고리즘을 사용했다.

DQN(Deep Q-Network)

DQN 알고리즘은 기존의 Q-learing과 딥러닝을 결합시킨 것이다. 기존의 Q-learning은 state-action 쌍에 해당하는 Q-value를 Q-table에 테이블 형식으로 저장하고 에이전트가 해당 state에서 action을 선택할 때 Q-table을 참고해 Q-value가 가장 큰 행동을 선택하는 방식이지만 이는 큰 문제점이 있다. 바로 state space와 action space가 커지게 되면 Q-table에 저장해야 할 값들도 많아지기 때문에 많은 메모리와 탬색 시간이 필요하게 된다. 이러한 문제점을 하기 위해서 딥러닝을 사용한다. 딥러닝을 이용해서 Q-value를 따로 저장하지 않고 Q-function을 근사시켜 Q-value를 구할 수 있게 된다.

DQN 알고리즘의 핵심 요소

1. experience replay

replay memory를 이용해서 랜덤하게 추출된 샘플들은 각각 다른 시간에서 수행된 샘플이므로 샘플들간의 연관성이 작기 때문에 Deep Q-learning의 sample correlation 문제를 해결 할 수 있다.

1. 매 스텝마다 추출된 샘플 e^t = (s_t , a_t , r_t , s_{t+1}) 을 replay memory에 저장함.

2. Replay memory에 저장된 샘플들을 랜덤하게 추출하여 학습에 이용함.

2. target network

기존의 Q-network를 동일하게 복제하여 main Q-network와 target network의 이중화 구조로 만들어 target network를 이용해 main Q-network를 업데이트 할 때 움직이는 target value로 인한 학습 불안정성을 개선한다.

$$\min_{\theta} \sum_{t=0}^{T} \left[Q(s_t, a_t; \theta) - \left(r_t + \gamma \max_{a'} Q\left(s_{t+1}, a'; \overline{\theta}\right) \right) \right]^2$$

1. Target network $\overline{\theta}$를 이용하여 target value $y_t = r_t + \gamma \max_{a'} Q\left(s_{t+1}, a'; \overline{\theta}\right)$를 계산함.

2. Main Q-network θ를 이용하여 action value $Q(s_t, a_t; \theta)$를 계산함.

3. Loss function $\left(y_t - Q(s_t, a_t; \theta)\right)^2$이 최소화되도록 main Q-network θ 업데이트 함.

4. 매 n스텝마다 target network $\overline{\theta}$를 main Q-network θ로 업데이트 함.

A2C(Advantage Actor-Critic)

A2C 알고리즘은 Actor-Critic 알고리즘을 기반으로 한다. Actor-Critic 알고리즘은 Actor network와 Critic network 2개의 네트워크(신경망)를 사용해서 학습을 진행한다. Actor는 정책(policy)을 학습하고 Critic은 주어진 정책의 가치를 평가하는 역할을 한다. A2C 알고리즘은 Actor-Critic 알고리즘에서 Actor의 기대 출력을 Advantage를 활용해 계산하는 것이다. Advantage는 현재 state-action 쌍의 가치와 평균 가치를 비교해서 얻어지는 것으로 양의 Advantage는 좋은 행동을 의미하고, 음의 Advantage는 나쁜 행동을 의미한다. 이렇게 A2C 알고리즘은 Advantage를 사용해서 Actor가 더 높은 확률로 더 높은 Advantage를 가지는 행동을 선택하도록 한다.

3.2.2. 강화학습 환경

강화학습 모델을 개발하기 위해서는 에이전트를 학습시킬 환경이 필요하다. 환경은 Open AI에서 제공하는 강화학습 환경개발 라이브러리인 Open AI Gym을 이용하여 구현하였다.

환경은 에이전트에게 현재 상태(State)를 제공한다. 에이전트는 환경으로부터 제공받은 상태를 통해 현재 상황을 이해하고, 적절한 행동(Action)을 결정한다. 에이전트로부터 행동을 받은 환경은 행동에 대한 보상을 에이전트에게 반환한다. 이 보상은 에이전트가 원하는 목표를 달성하는 데 얼마나 성공했는지를 나타내며, 에이전트는 이 보상이 최대가 되도록 결정하게끔 패턴을 학습한다. 강화학습 환경을 개발하기 위해서는 에이전트가 관측할 수 있는 상태, 에이전트가 결정할 수 있는 행동, 행동에 대한 보상이 구현되어야 한다.

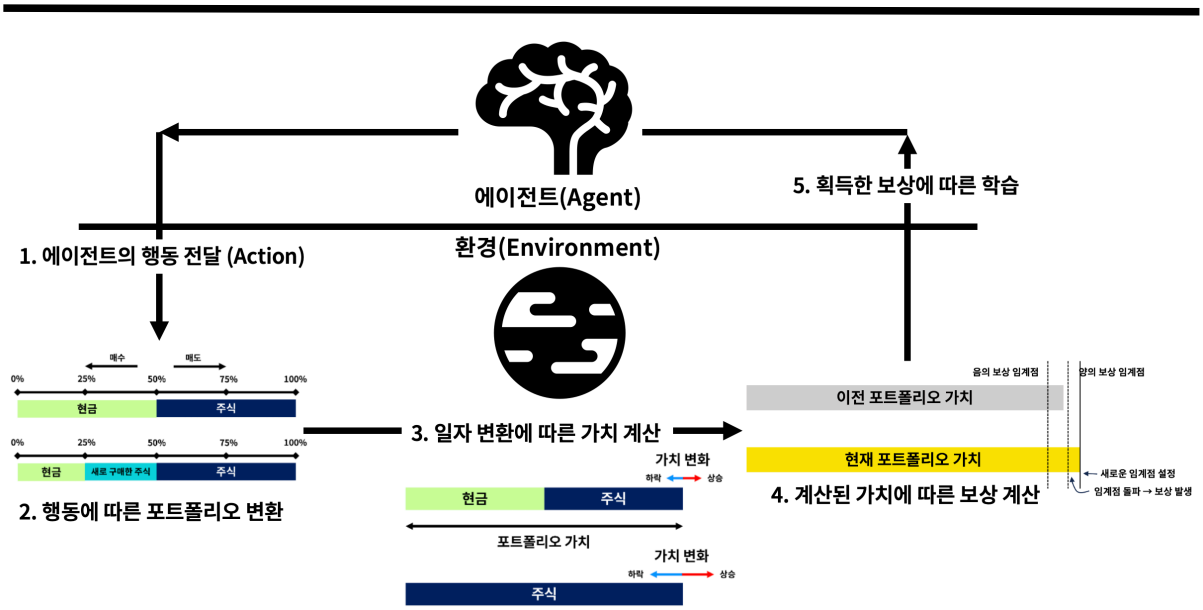


Figure 4. 강화학습 환경 흐름도

3.2.2.1. 행동(Action)

에이전트의 행동은 특정 상태에 대한 결정이며, 이를 통해 환경과 상호작용한다. 이 프로젝트에서는 두 가지 유형의 행동을 정의하고 이에 기반한 두 가지 환경인 BS(Buy and Sell) 환경과 LS(Long and Short) 환경을 구현했다. BS환경의 행동은 실제 거래를 나타내며 매수, 매도, 관망으로 구성된다. 반면 LS환경의 행동은 주가의 상승 또는 하락을 직접 예측하는 롱(Long)과 숏(Short)으로 구성된다. 이러한 두 가지 모델을 기반으로 성능을 비교했다. 실제 주식시장에서 long은 상승하는 주가에서 이득을 보는 투자전략을 의미하고, short은 하락하는 주가에서 이득을 보는 투자전략을 의미한다.

BS(Buy and Sell) 환경

BS환경은 가상의 초기 잔고를 가지고 에이전트의 행동에 따라 매수와 매도가 이루어지는 환경이다. 매수가 발생했을 때에는 주식의 보유 비중이 증가하고, 매도가 발생했을 때에는 보유 비중이 감소하며, 관망할 경우에는 현재 비중을 유지한다. 행동에 따라 변화하는 비중은 비율정밀도(proportion precision)라고 정의했다. 예를 들어 비율정밀도가 4인 경우 25%간격으로 비중 조절이 일어나고, 5인 경우 20%간격으로 비중 조절이 일어난다.

이 방법은 에이전트가 일관적이고 반복적인 행동을 선택할 때, 포트폴리오 비중이 의미있게 변화하고, 이에 따른 보상이 주어진다. 이를 통해 에이전트가 일관적인 행동을 취하도록 학습하기를 기대했다.

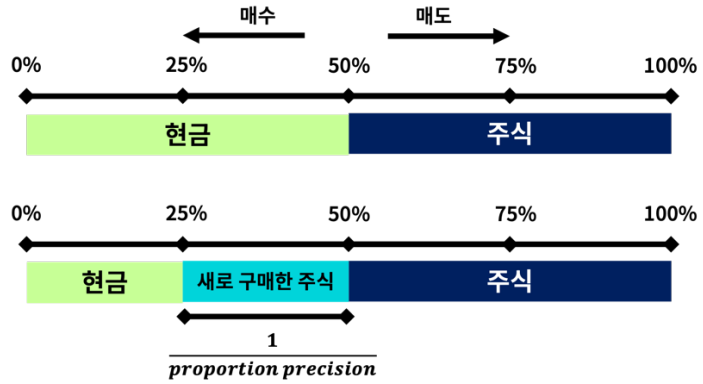


Figure 5. BS환경에서의 '매수' 행동 예시

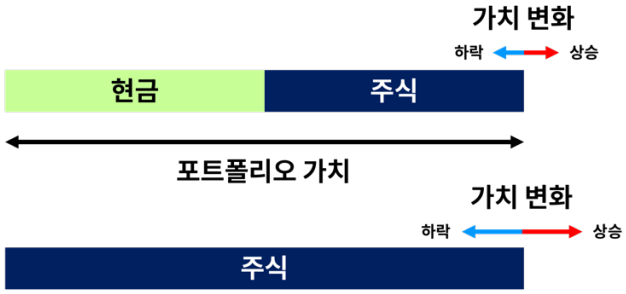


Figure 6.포트폴리오 비중에 따른 가치 변화, 주식의 비율이 클 수록 포트폴리오 가치가 크게 변한다.

또한 이와 같은 투자전략은 실제 주식 시장에서 '동일 비중 포트폴리오'이라는 이름으로 불리며 주가의 변동에 따라 현금 비중을 유지하고 조절하는 방법으로, 상승장 및 하락장에 효과적으로 대응할 수 있는 전략으로 알려져 있다. 이 전략을 적용함으로써 에이전트가 높은 수익률을 얻을 수 있기를 기대했다.

그러나 하이퍼파라미터를 조정하며 BS환경으로 생성한 일부 모델에서, 에이전트가 지나치게 관망을 선택하는 문제가 발생했다. 이러한 문제를 방지하고자 관망이라는 행동을 배제하고, 환경을 간단하게 설계하여 LS모델을 개발했으나, 후술할 LS환경의 문제점으로 인해 폐기하고, BS환경에서 학습한 에이전트를 이용한 모델을 과제에서 사용했다.

LS(Long and Short) 환경

LS환경은 에이전트가 주가의 변동을 예측하고 예측의 성공 정도에 따라 보상을 받는 환경이다. 에이전트는 롱(Long)과 숏(Short)을 행동으로 선택할 수 있다. 롱을 선택하고 주가가 상승하면 양의 보상을 받고, 숏을 선택했을 경우에는 주가가 하락함에 따라 양의 보상을 받는다. 예측이 빗나갈 경우 음의 보상을 받으며, 주가의 변화가 임계치를 넘지 못할 경우에는 보상을 받지 않는다.

LS환경은 오캄의 면도날(Ockham's razor) 이론에 기반하여 필수적 요소만을 남기고 설

계한 환경이다. 환경을 간단하게 작성함으로써 에이전트가 효율적이면서도, 안정적으로 학습할 수 있기를 기대했다. 더불어 상승예측(Long) 및 하락예측(Short)만을 행동으로 설정함으로써, BS모델에서 발생한 지나친 관망을 방지하고자 했다.

그러나 LS환경은 BS모델보다 더 심각한 편향된 행동을 유발했으며, 또한, 에이전트의 의사결정이 일관적이지 못하고 자주 변하는 경향이 있었다. 이를 방지하기 위해 보상의 scaling을 달리하거나, 잦은 변동에 대한 패널티를 부여했으나 유의미하게 개선되지 않았다. 이러한 불안정적인 모델의 결과는 사용자에게 제공하기 부적절하다고 판단하여 성능을 확인한 후 폐기되었다.

3.2.2.2. 상태(State)

상태는 환경이 에이전트에게 전달하는 상황에 대한 데이터이다. 에이전트는 환경으로부터 전달받은 상태와 학습한 패턴을 기반으로 행동을 결정한다. 과제의 환경에서는 일반적인 투자자가 관측을 통해서 획득할 수 있는 정보를 상태로 설정했다. 과제의 모든 환경은 현재 거래일로부터 window_size만큼의 데이터를 상태로서 제공한다. 크롤링과 API를 통해 추가적으로 획득한 지표들을 주가 데이터와 함께 에이전트에게 제공했다.

3.2.2.3. 보상(Reward)

성공적인 강화학습을 위해서는 에이전트가 자신의 행동에 대해서 평가할 수 있는 합리적인 보상이 설계되어야 한다. 주식투자를 학습하기 위한 보상 설계는 직관적이었다. 가장 기본적인 아이디어는 에이전트의 행동에 따라 투자를 진행하고, 주가가 변동함에 따라 포트폴리오의 가치가 변화함에 따라 적절한 보상을 지급하는 것이다.

그러나 주식 시장은 일반적으로 상승과 하락을 반복하면서 일정한 방향으로 일관되게 움직이지 않는다. 이러한 근시안적인 변동이 보상에 영향을 미칠 경우, 에이전트의 학습 효과를 저해할 수 있다고 판단했다. 또한, 우리는 에이전트가 단기적인 변동에만 주목하는 것이 아니라 시장의 큰 흐름을 파악하고 안정된 신호를 통해 투자하는 데에 집중하길 기대했다. 때문에 이번 과제에서는 보상 임계점을 설정함으로써 에이전트가 시장의 의미 없는 작은 변동에 혼란을 겪지 않고, 유의미한 신호에만 반응할 수 있도록 유도했다.

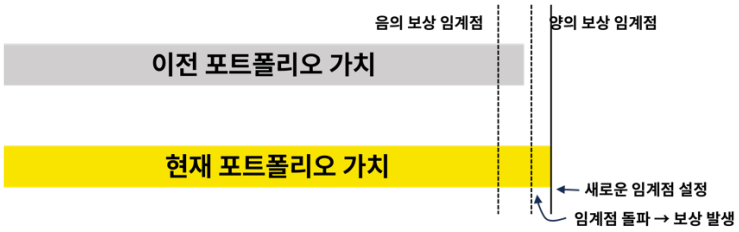


Figure 7. 보상 임계점을 설정하여 임계점을 돌파한 경우에 보상을 지급한다.

설계한 환경에서는 포트폴리오의 가치가 보상 임계점 이상으로 변화했을 때에만, 임계점을초과한 가치에 따라 보상이 주어진다. 하지만 보상 임계점을 돌파하지 못했을 경우에는 보상이 주어지지 않는다. 정확한 보상에 대한 식은 아래와 같다.

$$reward = \begin{cases} 0, & if\ (v_{current} - v_{last} < v_{last} * t) \\ \dfrac{v_{current} - v_{last}}{v_{last} * t}, & if\ (v_{current} - v_{last} \geq v_{last} * t) \end{cases}$$

$v_{current}$: *금일 포트폴리오 가치*, v_{last} : *직전 거래일 포트폴리오 가치*, t : *보상임계점*

예를 들어 보상 임계점을 3%로 설정했을 때, 초기 포트폴리오의 가치가 ₩10,000에서 시간이 흘러 ₩10,330으로 변동됐을 경우 (₩10,330 - ₩10,000) / (₩10,000 * 0.03) = 1.1이라는 보상이 지급되는 식이다.

3.3. 결과 제공을 위한 프론트엔드 개발

강화학습을 이용한 모델의 예측 결과를 사용자가 확인할 수 있도록 웹 인터페이스를 개발했다. 웹 인터페이스는 React를 이용하였다. 개발한 웹 페이지는 각 모델들에 대한 현재 주가와 강화학습 모델의 예측을 표시한다.

3.3.1. Mocking과 API specification

백엔드와 프론트엔드가 동시에 개발됨에 따라 완성된 API가 없는 상태에서 웹 사이트를 구축해야 했다. Mocking Service Worker를 이용하여, 가상의 데이터를 이용하여 웹 페이지의 작동을 확인하며 개발했다. 또한 개발이 완료되었을 때, 백엔드와의 원활한 연동을 위하여 Swagger를 이용하여 API명세를 작성한 다음 개발을 시작했다.

Figure 8. swagger로 작성한 API명세

3.3.2. 메인 화면

메인 화면은 기본적으로 리스트 형식을 통해 현재 주가와, 등락률, 개발된 모델들의 예측을 표시했다. 검색창을 통해 사용자가 종목과 종목코드를 검색할 수 있도록 하였고, 열 제목을 클릭하여 특정 값에 따라 정렬되도록 구현했다. 모바일로 확인할 수 있도록 하고 싶었으나, 가로로 긴 수치를 세로로 긴 모바일 환경에서 이용하기에는 불편함이 있었다. 때문에 반응형 디자인이 적용된 카드를 설계함으로써 모바일에서 시각적으로 안정적인 인터페이스를 구현하였다.

Figure 9. 메인 화면 리스트

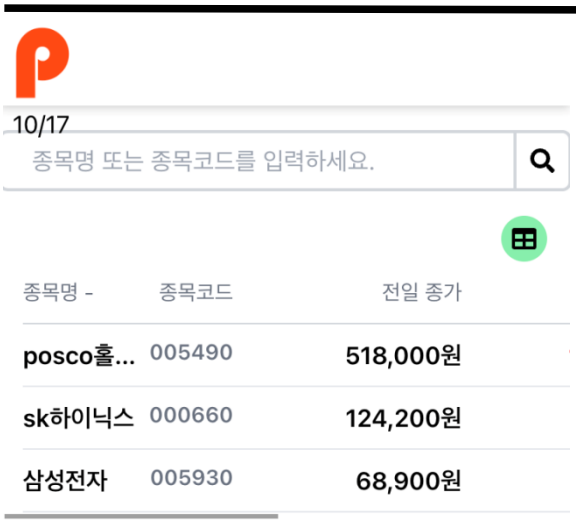


Figure 10. 모바일 환경에서의 표 형식

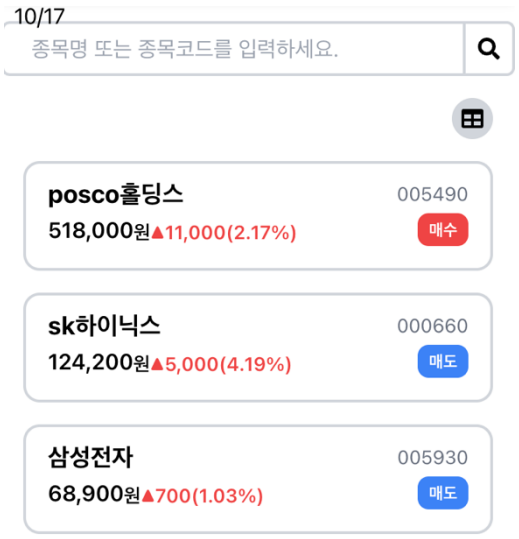


Figure 11. 카드 형식

3.3.3. 상세 화면

상세 화면에서는 거래일에 따른 주가의 변화와, 각 거래일에 따른 모델의 예측을 표시한다. 각 그래프의 바(bar)에 커서를 올리면, 해당 거래일에 대한 정확한 가격과 모델의 예측을 툴팁 형식으로 표시한다. 이 툴팁은 모바일에서도 이용할 수 있도록 커서 뿐만 아니라 터치에서도 반응하도록 구현했다. 또한 그래프를 다양한 배율로 볼 수 있도록 확대와 축소 기능도 구현하였다. 아래는 mocking된 데이터를 이용화여 화면을 구성한 것이다.

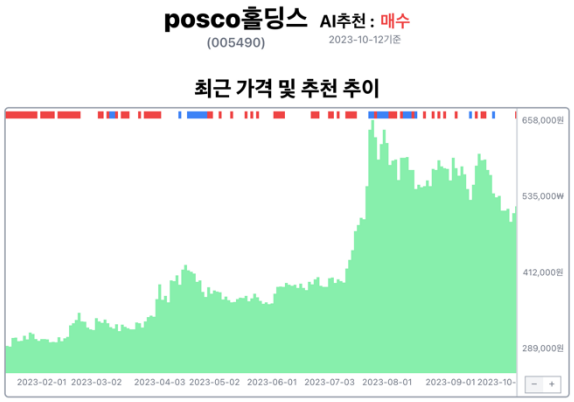


Figure 12. 그래프 축소 확대 기능

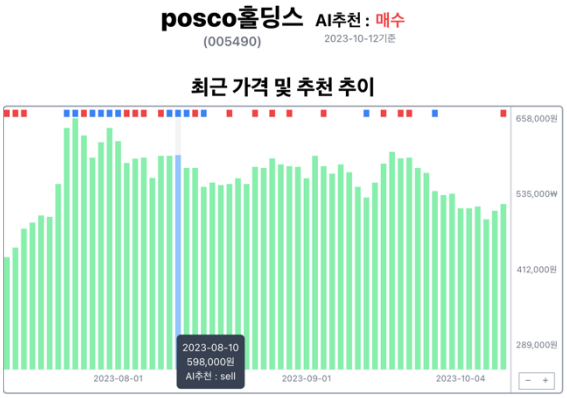


Figure 13. 마우스 커서 툴팁

3.3.4. 배포

과제 초반, 배포를 위해 깃허브 페이지(github pages)를 이용하려고 했으나, 깃허브 페이지에는 동적 웹을 배포할 수 없었다. 때문에 파이어베이스(Firebase)를 이용해 배포했다.

파이어베이스는 구글에서 제공하는 모바일 및 웹 어플리케이션 개발 플랫폼으로, 파이어베이스의 호스팅(Hosting)서비스를 이용해 웹을 손쉽게 배포할 수 있다. 또한 파이어베이스와 깃허브 액션(Actions)을 연동하여 CI/CD(Continuous Integration/ Continuous Deployment)를 적용할 수 있었다. 이를 통해서 앱에 변동사항이 생길 경우 배포 전 오류를 미리 확인하고 끊김없이 서비스를 업데이트할 수 있다.

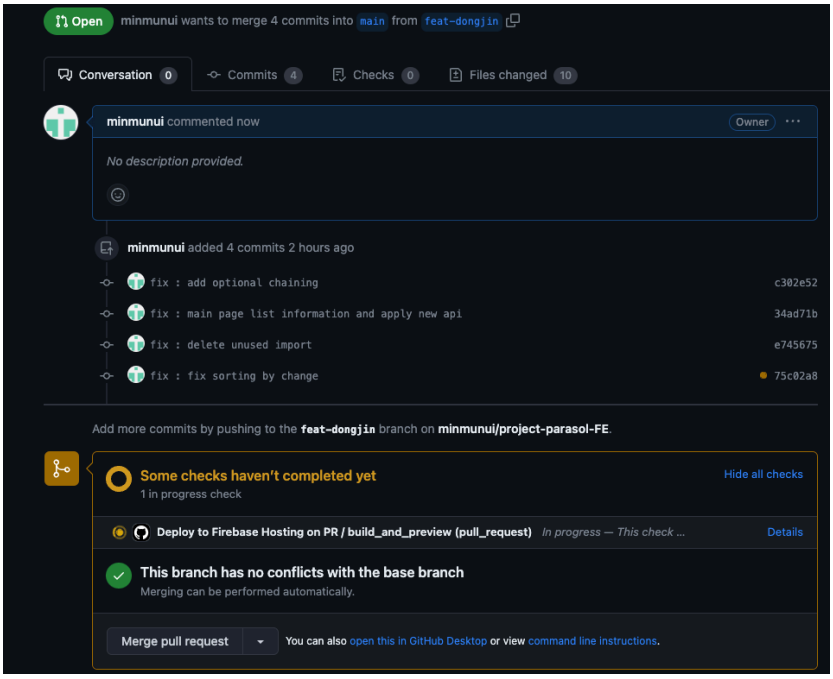


Figure 14. github PR에서 Merge이전 CI/CD를 위해 충돌을 자동적으로 확인한다

3.4. 데이터 저장/제공을 위한 백엔드 개발

강화학습 결과를 사용자들이 볼 수 있는 웹으로 가져오기 위하여 JAVA Spring Boot를 이용해 API 서버를 개발하였다. API End Point는 /stocks, /stock/{stockCode}, /recommend/{stockCode}/{page}, /prices/{stockCode}/{page}의 4가지가 있다. /stocks는 사용자가 웹에 처음으로 접속하였을 때 각 종목들에 대한 데이터를 한 눈에 볼 수 있도록 하기 위한 데이터들이 저장되어 있다.

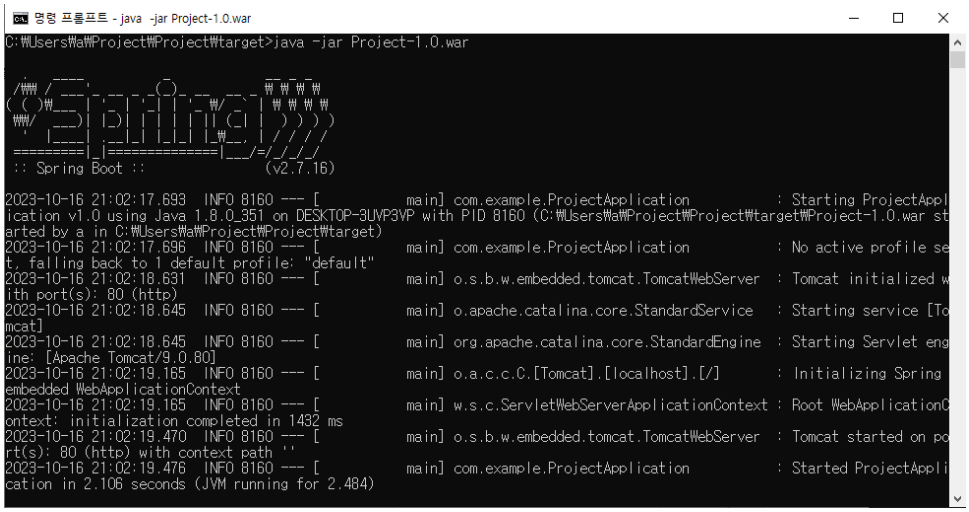


Figure 15. 백앤드 cmd 동작

4. 연구 결과 분석 및 평가

4.1. 평가 방법

모델의 성능을 평가할 때 수익률을 기준으로 평가했고 모든 모델에 동일하게 수수료(commission) 0.0003, 매도세(selling_tax) 0.00015를 적용했다.

학습 시에는 2014년 1월부터 2021년 12월까지 약 8년치의 데이터를 사용했고 테스트 시에는 2022년 1월부터 2023년 9월까지 약 21개월의 데이터를 사용했다.

주식 종목과 알고리즘에 따라 환경, 데이터 셋, 하이퍼파라미터(학습률, 버퍼 사이즈, 배치 사이즈, n steps 등)등의 조합을 달리하여 많은 모델을 생성한 후, 테스트를 거쳐 가장 수익률이 높은 모델들 중 안정성이 뛰어난 모델을 선정하여 최종적으로 사용하였다.

아래의 그래프들은 각 요인들에 대한 수익률 비교를 위한 그래프이며 최종적으로 사용한 모델의 수익률 및 액션은 5장 결론 부분에서 보일 것이다.

4.2. 데이터 셋에 따른 결과

본 과제에서는 2개의 주식 데이터 셋을 준비하고 학습에 사용하였다. A 데이터 셋은 ['Date', 'Close', '대비', '등락률', 'BPS', 'PBR', '주당배당금', '배당수익률'], B 데이터 셋은 ['Date', 'Close', '등락률', 'PER', 'PBR', '회전율', '환율', '코스피', '나스닥']을 사용했다.

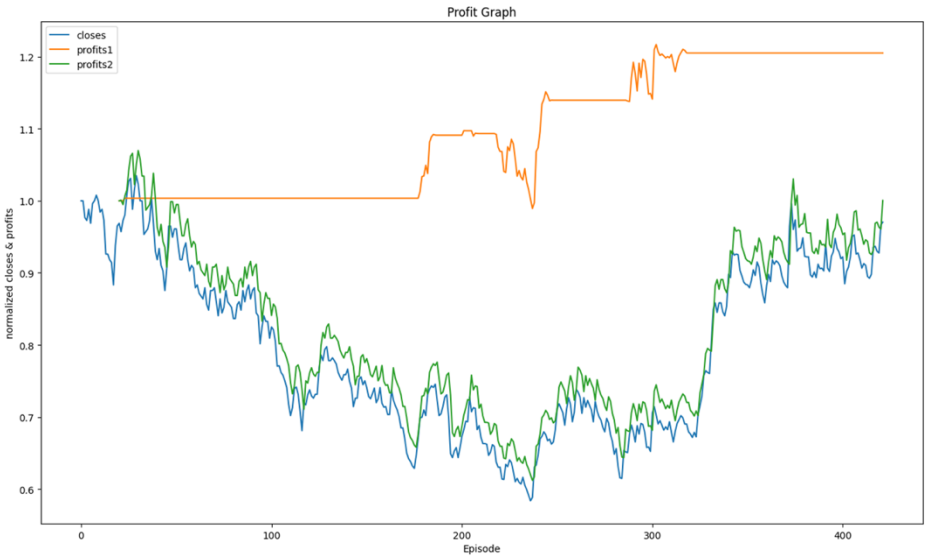


Figure 16. 데이터 셋에 따른 수익률 차이 그래프

파란색 그래프(closes)는 해당 주식 종목의 종가(close)를 나타내고 주황색 그래프(profits1)는 A 데이터 셋을 이용해서 학습을 진행한 모델의 수익률(profit rate)을 나타내고 초록색 그래프(profits2)는 B 데이터 셋을 이용해서 학습을 진행한 모델의 수익률을 나타낸다.

학습 및 테스트에서 주식 종목은 SK하이닉스, 환경은 BS(buy and sell) 환경, 강화학습 알고리즘은 DQN으로 일치시켰고 하이퍼파라미터(hyperparameter) 또한 학습률(learning rate) 0.0001, 리플레이 버퍼 사이즈(buffer size) 100K, 배치 사이즈(batch size) 512로 일치시켰다.

그래프의 개형을 보면 알 수 있듯이 A 데이터 셋을 사용해 학습을 진행했을 때는 약 20%의 수익률을 기록한 반면, B 데이터 셋을 사용해 학습을 진행했을 때는 종가 그래프를 거의 똑같이 따라 움직이는 것으로 보아 제대로 학습이 이루어지지 않은 것을 볼 수 있었다. 이는 에이전트가 초반에 buy 액션을 return하고 이후에 sell 액션을 return하지 않았음을 의미한다.

4.3. 강화학습 알고리즘에 따른 결과

본 과제에서는 에이전트를 학습시킬 때 DQN 알고리즘과 A2C 알고리즘을 사용하였다.

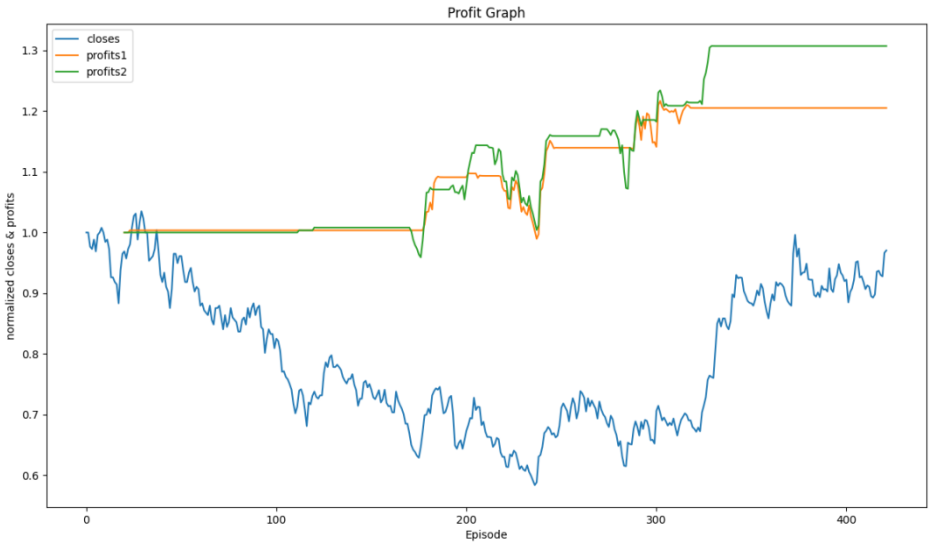


Figure 17. 강화학습 알고리즘에 따른 수익률 차이 그래프

마찬가지로 파란색 그래프(closes)는 해당 주식 종목의 종가(close)를 나타내고 주황색 그래프(profits1)는 DQN 알고리즘을 이용해서 학습을 진행한 모델의 수익률(profit rate)을 나타내고 초록색 그래프(profits2)는 A2C 알고리즘을 이용해서 학습을 진행한 모델의 수익률을 나타낸다.

학습 및 테스트에서 주식 종목은 SK하이닉스, 환경은 BS(buy and sell) 환경으로 일치시켰고 하이퍼파라미터(hyperparameter)의 경우에는 DQN 알고리즘과 A2C 알고리즘에서 사용하는 값이 다르기 때문에 일치시키지 못 했다.

그래프의 개형을 보면 알 수 있듯이 A2C 알고리즘을 사용해서 학습시킨 모델은 약 30%의 수익률을 기록했고 DQN 알고리즘을 사용해서 학습시킨 모델은 약 20%의 수익률을 기록했다. 특정 종목에 대한 결과이기 때문에 A2C 알고리즘이 DQN 알고리즘보다 더 높은 성능을 보인다고 단언할 수는 없지만 SK 하이닉스에 대해서는 A2C 알고리즘이 DQN 알고리즘보다 더 좋은 성능을 보였다.

4.4. 환경에 따른 결과

과제 진행에 있어, 에이전트를 학습시키는 환경이 가장 중요하다고 생각하여 환경 구축에 많은 시간을 투자했다. 여러 개의 환경을 구축했지만 크게는 BS(buy and sell) 환경과 LS(long and short) 환경으로 나눌 수 있다.

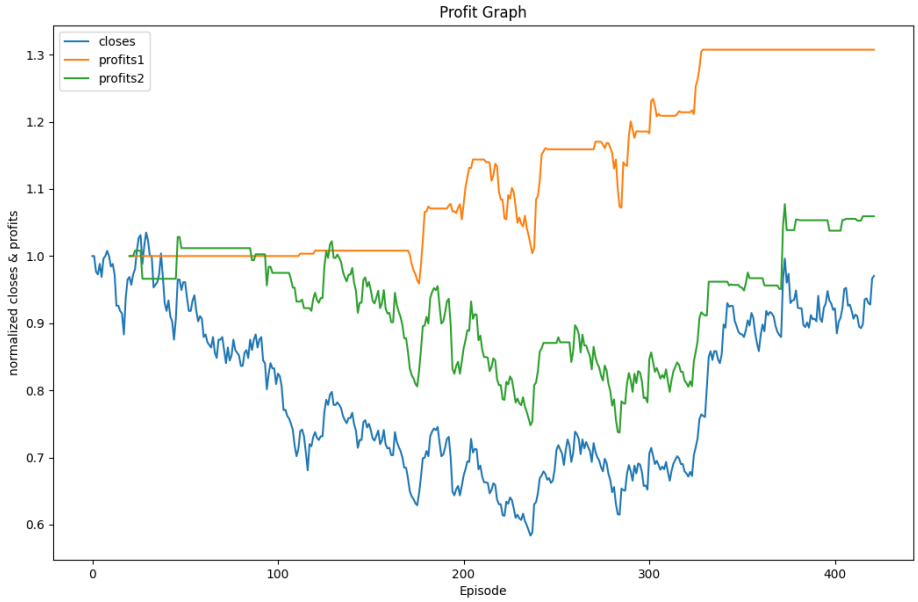


Figure 18. 환경에 따른 수익률 차이 그래프

마찬가지로 파란색 그래프(closes)는 해당 주식 종목의 종가(close)를 나타내고, 주황색 그래프(profits1)는 BS(buy and sell) 환경을 이용하여 학습을 진행한 모델의 수익률(profit rate)을 나타내고, 초록색 그래프(profits2)는 LS(long and short) 환경을 이용하여 학습을 진행한 모델의 수익률을 나타낸다.

학습 및 테스트에서 주식 종목은 S하이닉스, 강화학습 알고리즘은 A2C로 일치시켰다.

그래프의 개형을 보면 알 수 있듯이 BS 환경을 이용해서 학습시킨 모델은 약 30%의 수익률을 기록했고 DQN 알고리즘을 사용해서 학습시킨 모델은 약 5%의 수익률을 기록했다. 하이퍼파라미터, 데이터 셋, 강화학습 알고리즘 등 많은 요인에 따른 모델의 수익률을 비교하기 위해 수많은 학습 및 테스트 과정을 진행했지만 거의 모든 경우에 BS 환경에서 학습된 모델이 LS 환경에서 학습된 모델보다 높은 수익률을 기록하며 더 좋은 성능을 보였다.

5. 결론 및 향후 연구 방향

최종적으로 가장 잘 학습이 됐다고 생각하는 모델의 수익률 그래프는 아래와 같다. 아래와 같은 성능을 내는 모델들의 예측들을 배포된 웹을 통해 제공한다. 파란색 그래프는 해당 주식 종목의 종가(close)를 나타내고, 주황색 그래프는 학습된 모델이 해당 주식 종목에 대해 투자를 진행했을 때 얻은 수익률(profits)을 나타낸다.

종가 그래프(파란색 꺾은선 그래프) 위의 점은 학습된 모델의 액션을 나타내는데 파란색 점은 매도(sell)를 의미하고 빨간색 점은 매수(buy)를 의미한다. 점이 없는 지점은 관

망(hold)을 의미한다.

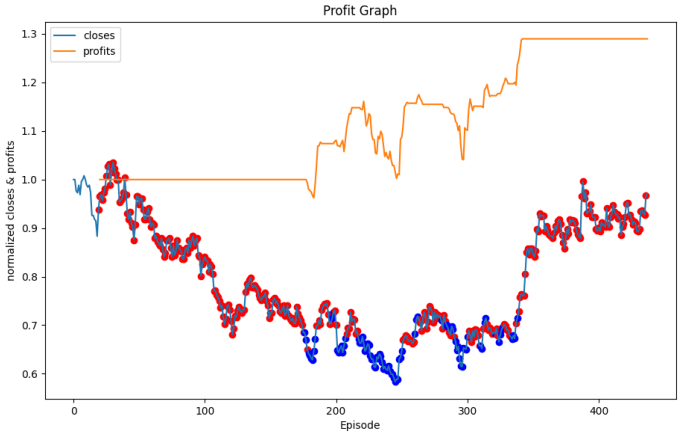


Figure 19. SK하이닉스 종목을 A2C로 학습한 모델의 수익률 그래프

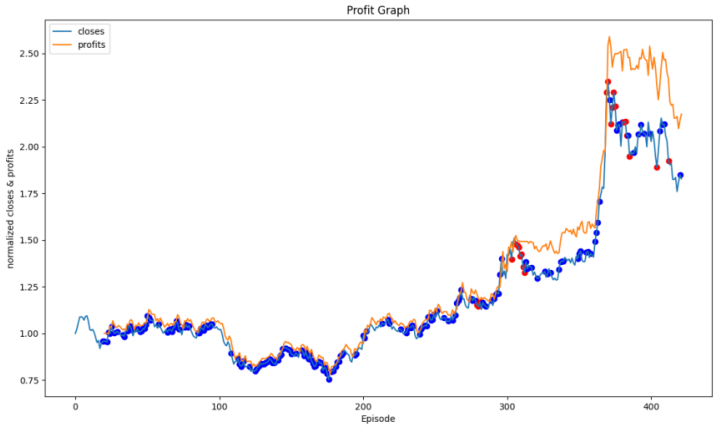


Figure 20. POSCO홀딩스를 DQN으로 학습한 모델의 수익률

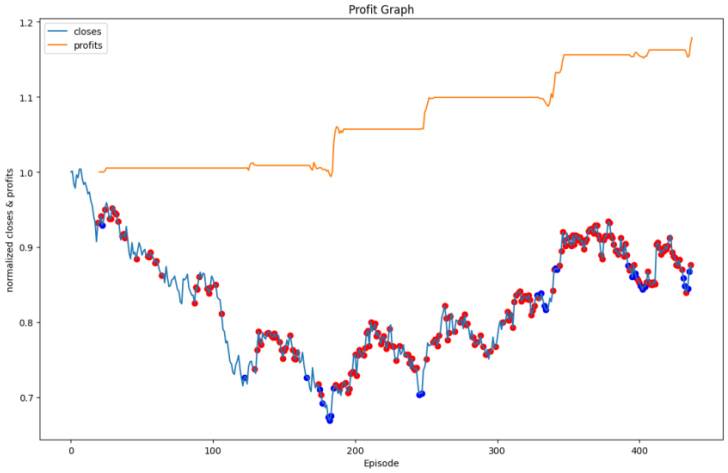


Figure 21. 삼성전자를 DQN으로 학습한 모델의 수익률

5.1. 결론

위에서 학습된 모델이 기록한 수익률 그래프를 보면 알 수 있듯이 강화학습을 이용해

학습시킨 모델은 3개의 주식 종목(SK하이닉스, POSCO홀딩스, 삼성전자)에서 주식을 사고 팔아야 할 적절한 타이밍을 예측하여 상승장에서 이득을 보는 것은 물론이고 하락장에서 손실 최소화하여 이를 바탕으로 약 20% 이상의 수익률을 기록했다. 따라서 주식 매매 전략을 세우기 위해 주식 환경에 강화학습을 적용하는 것은 꽤 유의미한 성과를 낸다는 것을 알 수 있다.

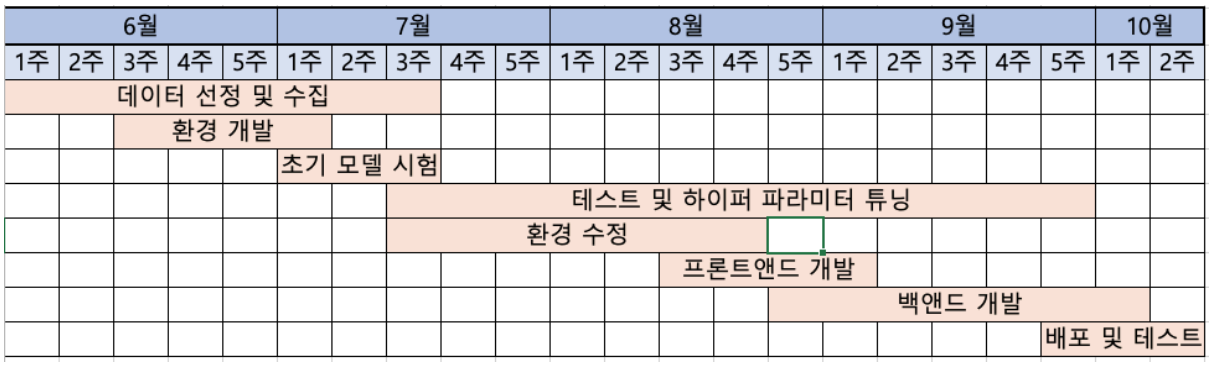
5.2. 연구 방향

수많은 주식 종목 가운데 이 프로젝트에서는 3개의 주식 종목만을 다루었기 때문에 일부 차트 패턴에서만 강화학습을 적용해본 셈이고 따라서 모든 주식 환경에서 강화학습이 매번 좋은 결과를 낸다고 단언하기엔 이르다. 따라서 더 많은 주식 종목에서 학습을 진행함으로써 강화학습 알고리즘, 주식 데이터 셋, 하이퍼파라미터 조합 등과 모델의 성능 간의 상관관계를 더 정확히 파악할 필요가 있다.

또한 강화학습 알고리즘을 DQN, A2C 이외에 A2C(soft actor-critic), DDQN(double DQN), DDPG(Deep Deterministic Policy Gradient) 등의 알고리즘을 적용해서 강화학습 알고리즘의 성능 개선을 기대할 수 있을 것이고, 에이전트의 학습을 위한 네트워크(신경망)을 DNN 이외에 LSTM, GRU 등의 시계열 데이터 처리에 좋은 성능을 보이는 네트워크를 적용할 경우 더 좋은 학습 결과를 기대할 수 있다.

6. 구성원 역할 및 개발 일정

정희영	학습된 모델 테스트, 하이퍼파라미터 튜닝
신재환	데이터 수집, 백앤드 개발
박동진	전체적인 프로젝트 구상, 강화학습 환경 개발, 프론트앤드 개발



7. 참고 문헌

권티랩. (2022). 파이썬을 이용한 딥러닝/강화학습 주식투작(개정2판) : 파이토치와 케라스를 활용한 인공지능 퀀트 투자 시스템. 위키북스

이용원, 양혁렬, 김건우, 이영무, 이의령. (2020) 파이썬과 케라스로 배우는 강화학습. 위키북스

전병조 & 노동건. (2021). 다양한 강화학습 기법을 이용한 업종별 주가예측 비교. 한국통신학회 학술대회논문집, 457-458

박정연, 홍승식, 박민규, 이현. (2021). 강화학습 기반 주식 투자 웹 서비스, 문화기술의 융합, 7(4), 807-814