심층강화학습을 이용한 최적 주식 거래 정책

김건, 김유빈, 홍인기
경희대학교
gun, dbqls15, ekhong@khu.ac.kr

Deep Reinforcement Learning for Optimal Stock Trading Policy

Geon Kim, Yu-Bin Kim, Een-Kee Hong
Kyunghee University

요 약

최근 OpenAI, 구글의 Deepmind 등 세계 여러 기업들에 의해 강화학습 알고리즘이 비약적으로 발전하고 있다. 이에 따라 인공지능 분야에서는, 국내외적으로 강화학습에 대한 관심이 커지고 있는 상황이다. 강화학습과 딥러닝 기법을 결합한 학습 방법을 심층강화학습(Deep Reinforcement Learning)이라고 한다. 본 논문에서는 심층강화학습의 알고리즘 중 하나인 DQN(Deep Q Learning)을 이용해 에이전트에게 주식 데이터를 학습시키고, 학습 이후 최적의 주식 매매 정책을 찾아내는 알고리즘을 구현하였다. 에이전트가 입력받는 학습 데이터는 각 날짜의 주가뿐만 아니라 보다 더 정확한 예측을 위해 당일의 거래량도 입력으로 추가하였다. 심층강화학습 알고리즘인 DQN(Deep Q Learning)을 이용해 구현한 Agent는 Benchmark Model보다 더 많은 이익을 낼 수 있었다. 후에 이 프로그램을 보다 정밀한 심층강화학습 알고리즘을 이용해 구현한다면 보다 더 많은 이익을 얻을 수 있는 주식 매매 알고리즘을 구현할 수 있을 것이라 기대한다.

Ⅰ. 서 론

본 논문에서 사용된 알고리즘을 간단하게 설명하겠다. 강화학습에서 사용한 에이전트(Agent)는 시작 시 일정량의 자본을 가지고 하루에 주식 2종류, A와 B를 한번 거래한다. 이 때 에이전트가 취할 수 있는 동작(Action)은 5가지이다. 'A 매수, A 매도, B 매수, B 매도, 아무것도 하지 않음' 총 5가지 동작을 Agent는 취하게 된다. 또한 문제를 간단하게 설계하기 위해 매수와 매도할 때 발생하는 수수료는 제외하였다.

본 논문에서 사용된 강화학습 알고리즘은 Deep Q Learning[1]이다. 인공 신경망은 복잡하게 구조화된 데이터에 대해 좋은 성능을 뽑아내는데 탁월하다. 상태(State)를 입력으로 받고 동작(Action)을 출력으로 뽑아내는 인공신경망으로 Q 함수를 나타낼 수 있다. 이 방법은 상태(State) 하나만을 입력으로 받고, 출력으로 나온 가능한 모든 동작(Action)들의 Q 값 중 가장 큰 값만 골라 Q 값을 업데이트할 수 있다는 장점이 있다. 이 연산을 통해 어떤 상태(State)에서든지 최적의 동작(Action)을 찾을 수 있다. 다음 그림은 앞에서 말한 동작을 간단하게 다이어그램으로 나타낸 것이다.

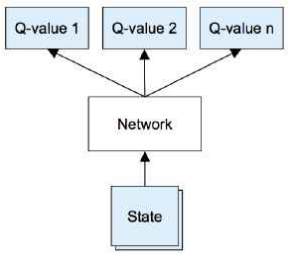

Figure 1.

본 알고리즘은 메모리에 <S(상태), A(동작), R(보상), S'(다음 상태)>를 tuple 형식으로 저장한다. 이를 이용해 Q-table 값을 업데이트 하게 되는데 그 순서는 다음과 같다.

1. 현재 상태를 Network에 입력하고 가능한 모든 액션에 대한 예측 Q 값을 가져온다.
2. 다음 상태를 Network에 입력하고 마찬가지로 가능한 모든 액션에 대한 예측 Q' 값을 가져온다.
3. 동작 a의 Q 값 target을 2번에서 구한 예측 Q'값 중 가장 큰 값으로 정한다. 다른 동작들의 Q 값 target은 1번에서 구한 예측 Q값으로 하여 error가 0이 되도록 한다.
4. 가중치를 역전파(Backpropagation)를 이용해 업데이트한다.

본 알고리즘의 성능을 비교하기 위해 Benchmark Model을 도입하였다. 비교 대상이 되는 Benchmark Model을 간단히 설명하자면 다음과 같다.

- 시작 : 주식 A와 B의 시작 자본의 절반만큼을 주식 구매에 사용한다.
- 기간의 10분의 1마다 보유한 주식의 10%를 매도한다, 따라서 현금보유량은 증가하게 된다.
- 자산 가치는 다음 식과 같다:
  - 자산 가치 = (주식 A 보유량 * A의 주가) + (주식 B 보유량 *B의 주가) + 현금보유량

위와 같은 값을 가지게 되는 Benchmark Model의 그래프와 본 논문에서 구현한 알고리즘의 그래프를 동시에 나타내어 한눈에 비교할 수 있게 나타낼 예정이다.

Ⅱ. 본론

본 논문에서 제안하는 알고리즘은 워렌 버핏과 같은 전문 투자자들을 대상으로 하는 것이 아니라, 평범한 일반 투자자들을 고려해 만들어졌다. 따라서 본 알고리즘은 주식 종목의 근본적이고 장기적인 트렌드를 보는 것이 아니라 규모가 작은 일일 변동량을 고려해 학습을 하게 된다. 또한 앞에서 설명했듯이 에이전트는 하루에 1주만을 거래가 가능하게 학습하였다.

■ State/Environment

본 알고리즘에 사용하는 주가는 오직 개장했을 때의 해당 주식의 가격이다. 또한 주식의 가격뿐만 아니라 에이전트는 필수적으로 다른 정보들을 필요로 한다. 에이전트가 필요로 하는 환경에 대한 정보들을 정리하면 다음과 같다.

- 매일 개장할 때 주식 가격
- 주식 A와 B의 일일 잔고량
- 에이전트가 주식을 살 수 있는 현금보유량
- 일일 자산보유량

이 네 가지 정보를 통해 Agent는 State를 평가하게 되고 그에 따른 정책을 만들어가게 된다.

본 알고리즘에서 학습을 하고자 하는 목표는 자산 가치를 최대화시키는 것이긴 하지만 단순히 자산 가치를 최대화 하는 것을 목표로 하고 있지 않다. 실제 일반 주식 거래자들은 주가가 하락할 때 심리적인 영향을 받고 이는 이후의 주식 거래에 영향을 준다. 실제 이러한 이유로 인해 주식을 낮은 가격에 매수 가장 고점에 매도하는 거래자가 매우 희귀하다. 본 알고리즘에서는 총 자산을 극대화시키는 것만을 목표로 하는 것이 아니라 자산의 일일 증가량을 최대화시키는 것도 목표로 하고 있다. 이는 알고리즘

의 신빙성을 올려주는 또 하나의 요인이다.

■ Agent

Agent를 정의할 때 사용한 파라미터들은 다음과 같다. 우선 Return을 계산하는데 사용되는 γ(감가율 :discount factor)의 값은 0.95로 설정하였다. 또한, 학습의 Exploration과 Exploitation을 결정하는 ϵ은 1에서 시작해 학습을 진행하며 0.01까지 줄어들게 만들었다. 이 때, ϵ는 에피소드가 한번 끝날 때마다 전 ϵ값에서 0.995배 줄어들게 된다. ϵ-greedy 방법을 채택함으로써 본 알고리즘은 Exploration과 Exploitation과 사이의 균형을 적절하게 맞추었다.

Agent에서 사용한 인공신경망은 앞에서 설명한 것과 같이 입력으로 state를 받고 출력으로 action을 뽑아낸다. 따라서 입력의 크기는 state_size와 같고 출력의 크기는 action_size와 같음을 알 수 있다. 인공신경망에 들어가는 입력과 출력을 정리하면 다음과 같다.

- 입력 : A의 주가, B의 주가, A의 잔고, B의 잔고, Agent 현금보유량, A의 5일 후 주가, B의 5일 후 주가
- 출력 : A 매수, B 매수, A 매도, B 매도, 아무것도 하지 않음

위의 항목에서 주목해야 할 것은 A와 B의 5일 후 주가도 입력으로 들어간다는 것이다. 그 이유는 Q Learning에서 사용하는 Bootstrapping 기법 때문이다. Bootstrapping이란 예측값을 이용해 또다른 값을 예측하는 것을 말한다. 본 논문에서 설명하고 있는 알고리즘은 여러 가지 시행착오를 거쳐 5일 뒤의 주가가 가장 안정적인 학습을 출력해낸다는 것을 알아냈다.

■ Deep Q Networks

인공신경망을 사용하며 발생하는 가장 큰 문제는 학습의 휘발성이다. 인공신경망은 한 step의 정보<S,A,R,S'>만을 받아들이기 때문에, 새로운 경험을 입력받게 된다면 그 전의 경험들을 망각하는 경향을 보인다. 이를 해결하기 위해 본 알고리즘은 Experience-replay[2]라는 기법을 사용하였다. Experience-replay기법은 "Replay buffer"를 추가로 구현한다. 이 buffer는 Agent가 Environment와 상호작용하며 만든 Experience tuple을 Batch의 크기만큼 저장하게 된다. 인공신경망은 추후에 저장된 Experience tuple을 랜덤으로 샘플링하여 강화학습에 이용해 똑똑한 학습을 하게 된다.[3] Experience-replay 기법을 이용하게 되면 학습의 휘발성이라는 문제점을 해결할 수 있게 된다.

DQN에서 인공신경망은 아래의 (1)식에 사용된다.[1]

$$Q(s,a) \approx r + \gamma * Q'(s',a') \quad (1)$$

- s : Agent's current state
- a : current optimal action
- γ : discount factor
- s' : next optimal state
- a' : optimal action in the next state

강화학습이 환경의 MDP에 대한 정보가 주어지지 않고 이루어지기 때문에 정확한 Q값을 알 수가 없다. 따라서 인공신경망을 이용해 Q(s, a)값을 근사하게 되는 것이다. 수많은 학습을 수행하면서 인공신경망이 근사한 Q(s, a)값은 결국 실제 Q값에 수렴하게 된다.

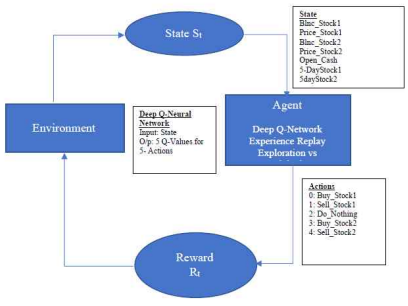

Figure 2.

위의 그림은 강화학습의 학습 전개 과정을 간단하게 블록 다이어그램으로 나타낸 것이다. 그림에서 설명된 순서로 Agent는 학습을 진행하게 되고 DQN 알고리즘을 통해 최적의 주식 거래 정책을 구하게 된다.

Ⅲ. 시뮬레이션 결과

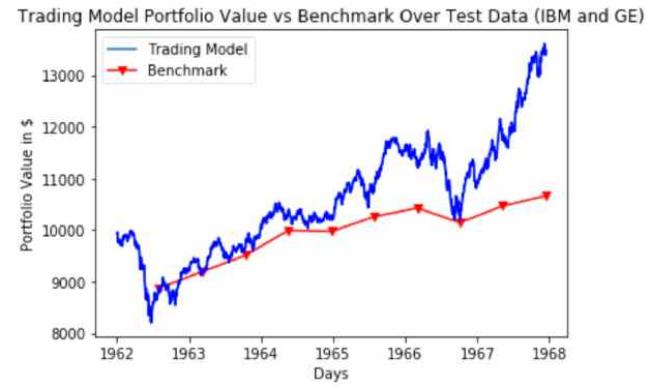

Figure 3.

1962년부터 1968년까지 IBM과 GE의 주식데이터를 가지고 본 알고리즘의 시뮬레이션을 돌려보았다. 주식 데이터는 Kaggle에서 제공되는 데이터를 활용하였으며[5], 앞에서 설명한 것과 같이 시장이 열렸을 때의 주가를 기준으로 학습하였다. 수치적으로 표현하자면 Trading Model의 총 자산이 Benchmark Model보다 30% 더 많다는 것을 결과로 알 수 있었다. 이는 본 알고리즘이 단순히 일정한 간격으로 보유한 주식의 10%를 매도한 Benchmark 모델보다 뛰어나다는 것을 증명하는 결과이다.

Ⅳ. 결론

본 논문에서는 DQN을 이용한 최적 주식 거래 정책을 구현하였다. 하지만 본 알고리즘은 한계점도 가지고 있다. 대표적인 것이 데이터 부족이다. 과거의 데이터만을 사용해 알고리즘을 만들었기 때문에 본 알고리즘이 현대의 주식시장에서 성능을 낼 수 있을지는 확신할 수 없다. 또한 주식시장의 개장 시간 때의 주가를 기준으로 알고리즘을 만들었기 때문에 주가에 대한 보다 더 정교한 단위가 필요하다. 예를 들자면 Meta에서 만든 FinRL에서 사용하였던 주가 단위가 있겠다.[5] 마지막 한계점은 DQN의 한계이다. Double DQN[4]등과 같은 DQN보다 더 강화학습의 단점이 보완된 알고리즘을 학습에 사용하게 된다면 보다 더 최적의 주식 거래 정책을 구할 수 있을 것이다. 따라서 다음 연구에서는 최근에 나온 강화학습 알고리즘을 이용해 본 알고리즘을 업데이트 해볼 예정이다.

참 고 문 헌

[1] Mnih et al., "Playing Atari with Deep Reinforcement Learning", arXiv:1312.5602 [cs.LG], December 2013. [Online].Available: https://arxiv.org/abs/1312.5602

[2] Matthew Hausknecht, Peter Stone, "Deep Recurrent Q-Learning for Partially Observable MDPs", arXiv:1507.06527 [cs.LG], January 2017. [Online].Available: https://arxiv.org/abs/1507.06527

[3] Hassel et al., David Silver, "Deep Reinforcement Learning with Double Q-learning",arXiv:1509.06461 [cs.LG], December 2015. [Online].Available: https://arxiv.org/abs/1509.06461

[4] Huang, Chien-Yi, "Financial Trading as a Game: A Deep Reinforcement Learning Approach", arXiv:1807.02787 [cx.LG], July 2018. [Online].Available: https://arxiv.org/abs/1807.02787

[5] Xiao-Yang Liul et al.,"FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance", arXiv:2011.09607 [q-fin.TR] March 2022. [Online].Available: https://arxiv.org/abs/2011.09607