선물 시장에서의 최적 스윙 탐색
: 강화학습 에이전트의 전략 학습

Learning the Futures Market's Flow

: A Reinforcement Learning Agent Mastering the Art of the Swing

YOLO(You Only Lose Once) 팀

이지민, 이수미, 이승연

이화여자대학교 통계학과

GITHUB

https://github.com/KanghwaSisters/YOLO-Futures.git

[초록]

최근 글로벌 금융시장은 코로나19 팬데믹, 지정학적 불안정성, 중앙은행 정책 변화 등으로 인해 높은 변동성을 보이고 있다. 이러한 환경에서 전통적인 기술적 분석이나 정량적 모델의 예측력은 한계를 드러내고 있으며, 이에 따라 불확실하고 동적인 시장 환경에서 연속적인 의사결정을 학습할 수 있는 강화학습(Reinforcement Learning, RL)이 금융 트레이딩 분야에서 주목받고 있다.

본 연구는 한국 KOSPI200 미니 선물의 실제 분봉 시계열 데이터를 활용하여 강화학습 기반 트레이딩 모델의 성능을 평가한다. 구체적으로, PPO(Proximal Policy Optimization) 알고리즘을 기반으로 한 에이전트가 KOSPI200 미니 선물 시장에서 수익성과 안정성을 동시에 확보할 수 있는지를 검토한다. 이를 위해 DLinear, CNN+Transformer, Informer 등 서로 다른 신경망 구조를 적용하여 금융 시계열 데이터의 학습 효율성을 비교하고, 누적 수익률과 Sharpe Ratio 등 위험 조정 성과 지표를 극대화하기 위한 다양한 보상함수를 설계하여 그 효과를 분석한다. 또한 OCHLV 분봉 데이터를 사용해 고빈도의 의사결정 상황을 모사하고, 슬리피지와 수수료 등 현실적인 거래 비용을 반영해 백테스팅의 타당성을 높였다.

실험 결과, KL로 진입 방향 규제를 추가하는 CTTS 모델은 테스트 구간에서 **평균 수익률 : 6.5%, Sharpe Ratio : 0.8, 최대 낙폭(MDD) : -12%, 승률 : 48%를** 기록하여 변동성이 높은 선물 시장에서 긍정적인 수익성과 효율적인 위험 관리 능력을 동시에 보여줬다. Informer 기반 모델 또한 높은 수익률을 달성했으나 변동성이 상대적으로 크게 나타나 Sharpe Ratio에서 아쉬운 성능을 보였다. DLinear 기반 모델은 다른 두 모델에 비해서는 낮은 성능을 보였으나, 진입 방향을 규제하지 않는 모델에 비해 높은 성능을 보이며 KL Divergence를 통한 진입 방향 규제가 효과적임을 입증했다.

본 연구는 단순히 가격 방향을 예측하는 데 그치지 않고, 강화학습 에이전트를 통해 매수·매도 강도와 자산 운용 전략 자체를 학습하도록 설계하였다. 이는 기존의 예측 중심 연구가 가격 예측과 별도의 규칙 기반 운용이라는 이원화된 접근을 취한 것과 달리, 트레이딩 정책을 직접 학습하여 의사결정과 자산 운용을 통합한다는 점에서 중요한 차별성을 갖는다. 본 연구는 강화학습 기반 금융 트레이딩 전략의 성능과 한계를 다각적으로 검토함으로써, 실제 시장 적용 가능성을 높이고 향후 금융 AI 연구의 확장 가능성을 제시한다.

핵심 주제어: 강화학습, 선물 트레이딩, 스윙 전략, PPO, Transformer , CNN, Informer, Dlinear

목차

I. 서론 ····· 5

II. 배경지식 ····· 6

A. 선물시장

B. 강화학습

C. 선물 시장의 MDP

D. 시계열 데이터

III. 핵심이론 ····· 13

A. PPO

B. 신경망

IV. 데이터 분석 및 전처리 ····· 19

V. 구현 ····· 22

A. 구조도

B. Agent 구현

C. 환경 구현

D. State 설계

E. Reward, Done 설계

F. 신경망 설계

G. 학습 설계

VI. 결론 및 시사점 ····· 38

그림 • 표 목차

[표 1] 미니 코스피200 주가지수 선물 상품 스펙

[표 2] 시계열 선물 시장 상태 변수

[표 3] 에이전트 상태 변수

[표 4] 보너스 표

[표 5] 에피소드 상황 별 종료, 청산 기준 표

[표 6] 학습-테스트 데이터 타임라인

[표 7] 테스트 구간에서 P&L (%)

[그림 1] 환경, 에이전트 사이의 상호작용

[그림 2] Short, Hold, Long에 따른 상태 전이 그래프

[그림 3] DLinear 구조도

[그림 4] CTTS 모델 전체 구조도

[그림 5] Informer 모델 구조도

[그림 6] 코스피 200 미니 선물 월별 시간별 종가 추이 (2010-2020)

[그림 7] 전체 데이터의 시간 별 빈도

[그림 8] 전체 프로젝트 구조도

[그림 9] 단일 DLinear 구조도

[그림 10] Multi-Head DLinear 기반 Actor-Critic Network 구조도

[그림 11] CTTS 기반 Actor-Critic Network 구조도

[그림 12] Informer 기반 Actor-Critic Network 구조도

[그림 13] 학습 사이클 종류

[그림 14] 수익률 히트 맵

[그림 15] 모든 에피소드의 평균 Sharpe Ratio

[그림 16] 모든 에피소드의 평균 MDD / 최저 MDD

[그림 17] 모든 에피소드의 평균 승률 (%)

I. 서론

최근 글로벌 금융시장은 코로나19 팬데믹, 지정학적 불안정성, 중앙은행 정책 변화 등으로 인해 높은 변동성을 보이고 있다. 특히 한국의 KOSPI200 선물 시장은 1일 평균 거래대금이 10조원을 넘나드는 대형 시장임에도 불구하고, 기관투자자와 개인투자자 간의 정보 비대칭과 고빈도 거래의 확산으로 인해 전통적인 기술적 분석이나 정량적 모델의 예측력이 제한되고 있다. 이러한 상황에서 강화학습은 불확실하고 동적인 환경에서 연속적인 의사결정을 통해 수익을 추구할 수 있다는 점에서 금융 트레이딩 분야에서 주목받고 있다. 기존의 지도학습 방식이 과거 데이터의 패턴을 학습하여 미래를 예측하는 데 그친다면, 강화학습은 시장 상황에 따라 매수, 매도, 관망 등의 행동을 직접 학습하여 실제 트레이딩 환경에 적합한 접근법을 제공한다.

본 연구는 KOSPI200 미니 선물의 실제 분봉 시계열 데이터를 활용하여 강화학습 기반 트레이딩 모델의 성능을 평가하는 것을 목표로 한다. 구체적으로는 PPO(Proximal Policy Optimization) 알고리즘 기반 강화학습 에이전트가 KOSPI200 선물 시장에서 유의미한 수익률을 달성할 수 있는지, DLinear, CNN+Transformer, Informer 등 서로 다른 신경망 구조 중 어떤 것이 금융 시계열 데이터의 특성을 효과적으로 학습하는지, 누적 수익률, 위험 조정 수익률 등을 극대화하기 위해 다양한 형태의 보상함수를 직접 설계하고 그 영향력을 분석하는지, 그리고 강화학습 기반 트레이딩 전략의 위험 대비 수익률과 최대 낙폭 측면에서의 안정성은 어떠한 지를 검토한다.

기존의 강화학습 기반 금융 연구는 주로 미국 주식 · 선물 시장이나 암호화폐 시장을 대상으로 이루어져 왔다. 이에 비해 본 연구가 활용한 국내 KOSPI200 미니 선물은 세계적으로 높은 거래대금을 기록하는 시장임에도 불구하고 분석 사례가 상대적으로 적어 연구적 의의를 지닌다. 또한 OCHLV 분봉 데이터를 이용해 실제 트레이딩 환경에 가까운 고빈도의 의사결정 상황을 모사하고, 슬리피지와 수수료 등 현실적인 거래 비용과 제약을 반영하여 백테스팅의 타당성이 높다. 더 나아가 본 연구는 단순히 가격의 상승 · 하락 방향을 예측하는 데 그치지 않고, 강화학습 에이전트를 통해 매수 · 매도 강도와 자산 운용 전략 자체를 학습하도록 설계하였다. 이는 기존의 예측 중심 연구가 가격 예측과 별도의 규칙 기반 운용이라는 이원화된 접근을 취한 것과 달리, 트레이딩 정책을 직접 학습하여 의사결정과 자산 운용을 통합한다는 점에서 중요한 차별성을 갖는다.

II. 배경 지식

A. 선물 시장

선물(Futures contract, Futures)은 파생상품의 한 종류로 품질, 수량, 규격 등이 표준화 되어있는 상품 또는 금융자산을 미리 결정된 가격으로 미래 일정시점에 인도 • 인수할 것을 약정한 거래이다. 현물 거래와 달리 선물 거래는 실제 상품을 거래하지 않으며, 계약 자체가 거래의 대상이며 지정된 거래소에서만 거래할 수 있다. 이 프로젝트에서는 선물 중 거래 승수를 낮춰 개인 투자자의 부담을 덜어주는 **미니 코스피200 주가지수 선물**을 거래 대상으로 선택했다.

기본용어

기초 자산 (Asset) : 선물 거래의 대상인 '표준화된 상품 또는 금융자산'

만기일 (Maturity) : 기초자산의 인도가 약속된 날

미결제약정 (Open interest) : 투자자가 보유하고 있는 계약

포지션 (Position) : 투자자가 특정 선물 계약을 매수/매도한 상태. 매수한 경우 롱(long) 포지션, 매도한 경우 숏(short) 포지션을 취했다고 함

증거금 (Margins) : 선물 계약 이행을 위한 보증금

위탁 증거금 (Initial margin) : 거래 시 최초로 납부하는 증거금

유지 증거금 (Maintenance margin) : 포지션 유지를 위해 계좌에 최소한으로 유지해야하는 증거금 수준

마진콜 (Margin call) : 거래 중 발생하는 손실로 계좌의 증거금 수준이 유지 증거금 이하로 떨어졌을 때, 증권사가 투자자에게 추가 증거금을 채워 넣으라고 요구하는 것

일일정산 (Daily settlement) : 매일 장 종료 시의 가격에 따라 현재 투자자가 보유 중인 미결제약정에 대해 증거금 계좌의 변동이 이루어지는 것

포지션

선물은 상승을 기대하는 주식 거래와 달리, 하락장에서도 수익을 얻을 수 있다. 선물은 미래 시점의 가격을 예측해 현 시점에서 매수, 매도 포지션을 정한다. 만약 미래 시점에 가격이 오를거라 기대한다면 지금 사서 비싸게 파는 롱 포지션(매수)를 취한다. 반대로 가격이 떨어질거라 기대한다면 지금 팔아서 나중에 싸게 구입하는 숏 포지션(매도)를 취한다. 매수도 매도도 하지 않는 선택은 홀드이며, 이전의 포지션과 체결량이 변화하지 않는다. 만약 현재 갖고 있는 포지션을 청산하고 싶다면 반대 포지션을 취해 실행한다. 현재 체결된 계약이 10계약 롱 포지션일 때, 10계약 숏 포지션을 취해 현재 포지션을 청산할 수 있다.

레버리지 상품

선물은 실제 계약금보다 적은 현금으로 거래할 수 있는 레버리지(leverage) 상품이다. 레버러지 상품은 소액으로 고액의 계약을 체결할 수 있기 때문에, 투자자에게 큰 수익 기회를 제공하지만 동시에 높은 리스크도 수반한다. 선물 계약을 시작하기 위해 필요한 것이 바로 위탁증거금(Initial Margin)이다. 이는 선물 계약에 진입하기 위해 예치해야 하는 최소 보증금을 의미한다.예를 들어, 키움증권 기준 위탁증거금율이 10.5%이라면, 1,500만 원짜리 선물 계약을 체결하기 위해서는 단 157.5만원만 있으면 거래가 가능하다. 반면, 동일한 자금을 가지고 일반 주식에 투자할 경우 157.5만원어치만 매수할 수 있어 투자 규모가 상대적으로 작다. 레버리지 거래의 가장 큰 특징은 수익률이 실투자금 대비 훨씬 커진다는 점이다.

예를 들어 지수가 1% 상승했을 때 일반 주식 거래는 자산의 1%인 15,750원의 수익을 가져가지만, 선물 거래는 전체 계약금 기준 수익(15만 원)을 실투자금(157만 5천 원)으로 나눈 약 10.5%의 수익률을 얻을 수 있다. 하지만 이와 같은 구조는 손실에도 동일하게 적용된다. 지수가 1% 하락할 경우, 선물 거래자는 실투자금의 10.5%에 해당하는 손실을 감수해야 한다. 이러한 손실 리스크를 관리하기 위한 장치가 바로 유지증거금(Maintenance Margin)이다. 유지증거금은 포지션을 계속 보유하기 위해 필요한 최소 자본금으로, 보통 위탁증거금보다 낮게 설정된다. 키움 증권 기준 유지증거금율이 7%라면, 1,500만 원짜리 계약을 유지하려면 최소 105만 원 이상이 계좌에 있어야 한다. 만약 미실현 손실로 인해 계좌 자산이 유지증거금보다 낮아진다면, 마진콜(Margin Call)이 발생한다. 이 경우, 투자자는 빠른 시일 내에 부족한 금액을 추가로 입금하거나, 보유 포지션을 일부 청산해야 한다. 아무 조치를 취하지 않으면, 거래소가 자동으로 보유 포지션을 강제 청산하게 된다. 강제청산은 대개 시장이 급변하는 시점에 이루어지기 때문에, 최악의 타이밍에 손실이 확정될 가능성이 높다. 이는 결과적으로 투자자에게 더 큰 손실을 안길 수 있다.

만기일과 결제월

미래에 상품을 거래하기로 한 날은 만기일이고, 만기일이 있는 달은 결제월이다. 선물 시장에는 6개월 치 결제월이 상장되어 있으며, 7월 기준 네 개의 비분기월 (8,10,11,1월)과 분기월(9, 12월)로 구성된다. 각 결제월에 대한 만기일은 해당 달의 두 번째 목요일이다. 만기일은 계약이 이행되기로 정해놓은 날이기 때문에, 만기일까지 유지한 포지션이 있다면 만기일의 종가를 기준으로 강제 청산이 일어난다. 선물 상품은 현재 시점의 근월물 뿐만 아니라, 그 이후 결제월에 대한 향후 결제월물도 미리 거래할 수 있다. 예를 들어, 7월에 9월물은 물론 12월물 선물도 미리 체결이 가능하다. 이와 같은 구조를 이용해 만기가 다가올 때 기존 포지션을 청산하고 더 먼 결제월물에 같은 방향의 포지션을 다시 잡는 것을 롤오버(Roll-over)라고 한다. 하지만 우리가 다루는 데이터는 근월물 거래 기준이기 때문에 롤오버를 고려하지 않는다.

손익 계산: 선입선출

선물 거래의 손익 계산은 일반적으로 먼저 체결된 계약부터 순차적으로 청산되는 선입선출법을 따른다. 신규 계약 진입 시 가용 계좌에서 위탁 증거금이 차감된다. 하루의 장 종료 시까지 포지션을 보유하고 있다면 매일 장 마감 이후 일일 정산되어 일일정산가격에 따른 손익이 계좌에 반영된다. 계약 청산 시에는 청산한 만큼 계약 진입 시 납입한 위탁 증거금이 계좌로 반환되며, 일일정산으로 정산되지 않은 나머지 부분의 손익이 계좌에 반영된다. 만약 만기일까지 보유 중인 미결제약정이 존재한다면 만기일의 일일정산 가격으로 모든 계약이 강제 청산된다.

미니 코스피200 주가지수 선물

미니 코스피200 주가지수 선물(Mini KOSPI 200 Index Futures)는 대한민국의 코스피200 주가지수를 기초자산으로 하는 선물거래로, 거래 단위와 증거금이 코스피200 선물의 1/5 사이즈로 구성된 상품이다. 구체적인 상품 스펙은 거래소마다 약간의 차이가 있을 수 있으나 대부분 아래와 같다.

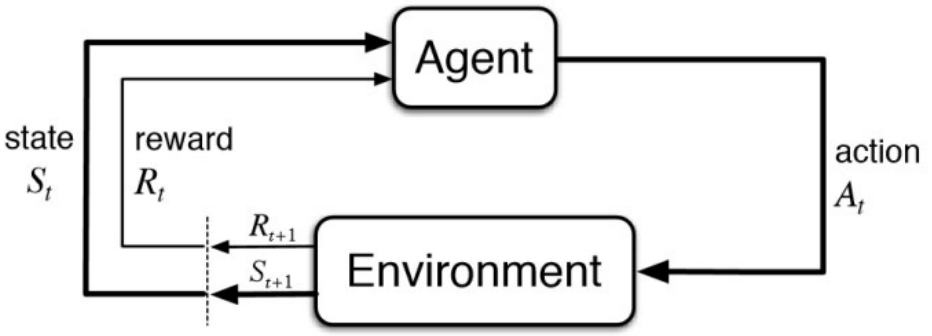
[표 1] 미니 코스피200 주가지수 선물 상품 스펙

거래 대상	코스피 200 지수
거래 단위	코스피 200 선물 pt * 5만(거래 승수)
결제월	매월
거래 승수	50,000원
호가 가격 단위 (tick)	0.02 포인트
최소 가격 변동	1000 원 (0.02 x 5만원)
거래 시간	08:45 – 15:45 (최종거래일 08:45 – 15:20)
최종 거래일	매월 두 번째 목요일 (공휴일인 경우 직전 거래일)
최종 결제일	최종거래일의 다음 거래일
위탁증거금	10.00% (2025.07.13 기준)
유지증거금	7.00% (2025.07.13 기준)
단일가격경쟁거래 (동일호가시간)	개장 시 08:30 – 08:45 거래 종료 시 15:35 – 15:45 (최종 거래일 15:20 – 15:45)

B. 강화학습

강화학습(Reinforcement Learning)은 지능형 에이전트(Agent)가 동적인 환경 (Environment)과 상호작용하며 누적 보상(Reward)을 극대화하는 정책 (Policy)을 학습하는 머신 러닝의 한 분야이다. 체스나 바둑 기사가 경기를 두는 과정과 유사하게, 에이전트는 주어진 상태(State)에서 어떤 행동(Action)을 취할지 결정하고, 그 결과로 환경으로부터 다음 상태와 함께 스칼라 형태의 피드백

신호, 즉 보상을 받는다.



[그림 1] 환경, 에이전트 사이의 상호작용

이 과정은 정답이 명시된 데이터를 기반으로 패턴을 학습하는 지도학습(Supervised Learning)이나 데이터의 내재된 구조를 파악하는 비지도학습(Unsupervised Learning)과 근본적인 차이를 보인다. 강화학습의 궁극적인 목표는 단기적인 보상이 아닌, 장기적인 관점에서 보상의 총합, 즉 누적 보상을 최대로 만드는 최적의 행동 전략을 발견하는 것이다.

이러한 목표를 달성하기 위해 에이전트는 탐험(Exploration)과 활용(Exploitation) 사이의 근본적인 딜레마에 직면한다. 활용은 현재까지의 경험을 바탕으로 가장 높은 보상을 기대할 수 있는 행동을 선택하는 것이고, 탐험은 더 나은 전략을 찾기 위해 불확실하더라도 새로운 행동을 시도하는 것이다. 과거에 유효했던 특정 매매 전략(활용)에만 의존할 경우, 시장의 구조적 변화(Regime Change)에 대응하지 못하고 큰 손실을 볼 수 있다. 따라서 장기적 최적해를 찾기 위해서는 미지의 가능성을 탐색하는 과정이 필수적이며, 이 딜레마를 효과적으로 관리하는 것이 강화학습 에이전트 설계의 핵심 과제이다.

C. 선물 시장의 MDP

선물 시장의 동적인 움직임을 모델링하기 위해서 Markov Decision Process (MDP)를 다음과 같이 정의한다.

s_t : 선물 시장의 가격 데이터와 현재 에이전트의 자산 상황

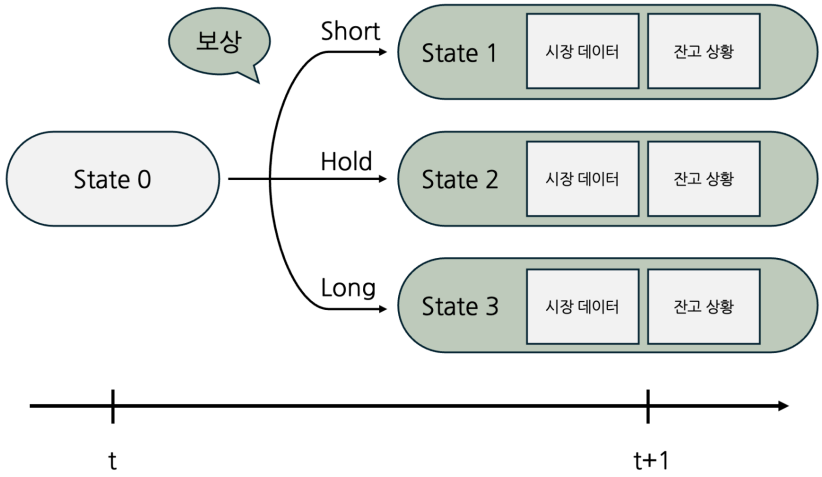
a_t : 숏, 홀드, 롱 포지션을 각각 몇 포지션까지 취할 것인지 (최대 10계약)

$r(s_t, a_t, s_{t+1})$: 현재 상태 s_t에서 새로운 상태 s_{t+1}로 전이될 때 취한 행동 a_t에 대한 보상

$\pi(s)$: 현재 상태 s에서 전체 행동 집합에 대한 확률

$V(s)$: 현재 상태 s의 가치

$Action\ Space$: {-k, …, −1, 0, 1, …, k} $s.t.$ k=10
행동의 절대값은 계약 수, 부호는 각기 숏(−)과 롱(+)을 의미. 0은 홀드



[그림 2] Short, Hold, Long에 따른 상태 전이 그래프

금융 시계열 데이터의 MDP는 어떤 행동을 선택하더라도 State가 전이된다는 특징이 있다. 에이전트가 Hold를 선택하더라도 시장가는 변하기 때문이다.

D. 시계열 데이터

시계열 데이터(Time Series Data)는 하나의 변수를 시간에 따라 여러 번 관측한 데이터로, 일정한 시간 간격으로 수집된 연속적인 관측 값을 의미한다. 금융 분야에서는 가격, 거래량, 변동성 등의 변화 예측과 반복되는 패턴에 대한 인사이트 도출을 목표로 한다.

시계열 데이터의 핵심 특성은 시간 t의 절대적인 순서가 중요하다는 점이다. 일반적인 횡단면 데이터와 달리, 과거의 값이 현재와 미래의 값에 영향을 미치는 시간적 의존성이 존재한다. 이러한 시간적 특성을 이해하는 것은 시계열 분석에서 매우 중요하다.

금융 시계열 데이터는 일반적인 시계열과 구별되는 독특한 특성들을 가진다. 첫 번째 특징은 높은 변동성(High Volatility)으로, 특히 변동성 군집화 현상이 나타난다. 이는 높은 변동성이 지속되는 기간과 낮은 변동성이 지속되는 기간이 구분되어 나타나는 특성이다. 두 번째 특징은 두꺼운 꼬리 분포(Fat Tail Distribution)로, 정규분포보다 극단적인 값들이 예상보다 자주 발생하는 현상이다. 이러한 현상은 금융 위기나 급격한 시장 변동 시 관찰된다. 세 번째 특징은 레버리지 효과(Leverage Effect)로, 가격 하락 시 변동성이 상승 시보다 더 크게 증가하는 비대칭적 특성을 나타낸다. 또한 대부분의 금융 시계열은 비정상성(Non-stationarity)을 가지며, 평균과 분산이 시간에 따라 변화한다. 효율적 시장가설에 따라 가격 변화는 예측하기 어려운 랜덤워크 특성을 보이기도 한다.

시계열 데이터의 변동을 이해하기 위해, 이를 네 가지 구성 요소로 나누어 분석할 수 있다. 추세(Trend)는 장기적인 증가 또는 감소 패턴을 나타내며, 계절성(Seasonality)은 특정 요일이나 계절에 따라 일정한 주기로 반복되는 패턴을 의미한다. 주기성(Cycle)은 고정된 빈도가 아니지만 형태적으로 유사하게 나타나는 패턴이며, 노이즈(Noise)는 측정 오류나 내부 변동성 등 다양한 요인으로 생기는 불규칙적인 변동이다. 이러한 구성 요소들은 가법 모델(시계열 = 추세 + 계절성 + 주기 + 노이즈) 또는 승법 모델(시계열 = 추세 × 계절성 × 주기 + 노이즈)로 결합되며, 시간에 따른 변동폭의 특성에 따라 적절한 모델을 선택한다.

시계열 데이터 분석에서는 자기상관(Autocorrelation)과 정상성(Stationarity)도 중요한 특성이다. 자기상관은 과거 값과 현재 값 사이의 상관관계를 나타내며, 시차(lag)가 짧을수록 높은 자기상관을 보이는 것이 일반적이다. 자기상관함수(ACF)를 통해 특정 시차별 자기상관 계수를 측정할 수 있으며, 부분자기상관함수(PACF)는 특정 시차에서의 직접적인 상관성을 측정하여 다른 시차들의 영향을 제거한 순수한 상관관계를 파악한다. 정상성은 시간에 따라 통계적 특성이 변하지 않는 경우를 의미하며, 평균, 분산, 자기공분산이 시간에 무관하게 일정해야 한다. 대부분의 금융 시계열은 추세나 계절성을 포함하여 비정상성을 보이는데, 이는 임의의 시점에서 얻은 모델을 미래에 적용할 수 없게 만들어 예측 성능을 저하시킨다. 정상성 검정은 ADF(Augmented Dickey-Fuller) 테스트나 KPSS 테스트 등의 단위근 검정을 통해 수행할 수 있다.

시계열 데이터를 분석하거나 모델에 활용하기 위해서는 적절한 전처리 과정이 필요하다. 결측치 처리에서는 전진 대체법(Forward Fill, ffill)이 데이터의 연속성을 유지하는 데 효과적이다. 정규화(Normalization)는 Min-Max 정규화나 Z-score 표준화를 통해 서로 다른 스케일의 변수들을 동일한 범위로 조정하며, 절대적 가격보다는 수익률이나 변화율로 변환하여 모델의 일반화 성능을 향상시킨다. 정상성 확보를 위해서는 차분(Differencing)이나 로그 변환 등을 사용하여 비정상 시계열을 정상화함으로써 모델 학습의 안정성을 높일 수 있다.

마지막으로, 강화학습 모델의 상태 설계에서 활용할 수 있는 기술적 지표를 살펴보면 다음과 같다.

기본 지표: 로그 수익률, 일정 기간의 수익률 계산, 롤링 윈도우를 통한 평균과 표준편차로 표준화 점수 계산

추세 지표: 서로 다른 기간의 지수이동평균(EMA) 교차, CCI(Commodity Channel Index), PSAR(Parabolic SAR)

모멘텀 지표: 스토캐스틱(%K, %D), ROC(Rate of Change), RSI(Relative Strength Index)

거래량 지표: OBV(On-Balance Volume), AD Line

변동성 지표: 볼린저 밴드 상·하한선, 밴드 폭, ATR(Average True Range), 갭 크기

금융 시계열, 특히 파생상품인 선물(Futures) 시장 데이터는 강화학습의 적용 가능성을 시험하기에 적합한 특성을 다수 내포하고 있다.

• 순차적 의사결정 문제 (Sequential Decision-Making)

트레이딩은 단일 예측 문제가 아니라 '진입, 청산, 관망'과 같은 일련의 행동들이 순차적으로 연결되어 최종 수익을 결정하는 문제이다. 이는 현재의 행동이 미래의 상태와 보상에 직접적인 영향을 미치는 강화학습의 마르코프 결정 과정(Markov Decision Process, MDP) 구조와 자연스럽게 부합한다.

• 극심한 비정상성(Non-stationarity)과 적응의 필요성

선물 시장 데이터는 평균, 분산과 같은 통계적 특성이 시간에 따라 끊임없이 변화하는 강한 비정상성을 보인다. 고정된 데이터셋으로 학습하는 전통적인 방식은 변화하는 시장 동역학을 따라가기 어렵다. 반면, 강화학습 에이전트는 환경과의 지속적인 상호작용을 통해 실시간으로 정책을 업데이트하므로, 변화하는 시장 상황에 적응(Adaptation)하는 능력을 내재적으로 학습할 수 있다.

• 명확한 보상 체계

트레이딩의 목표는 '수익 극대화'라는 명확한 지표로 귀결된다. 이는 강화학습의 '누적 보상 극대화' 목표와 직접적으로 연결될 수 있다. 실현 손익(Realized P&L)이나 샤프 지수(Sharpe Ratio) 등을 보상 함수로 직접 설계하여 에이전트의 학습 방향을 명확하게 유도할 수 있다.

• 모델 프리(Model-Free) 접근의 유효성

금융 시장의 복잡성과 무작위성을 정확한 수학적 모델로 정의하는 것은 거의 불가능에 가깝다. 모델 프리 강화학습 알고리즘은 시장의 작동 원리(모델)를 명시적으로 알지 못하더라도, 오직 경험(상태, 행동, 보상 샘플)만을 통해 최적 정책을 학습할 수 있어 이러한 환경에 효과적이다.

본 연구는 상기한 특성에 주목하여, 강화학습 알고리즘을 코스피 200 미니 선물 데이터 환경에 적용하고자 한다. 연구의 최종 목표는 변동성과 불확실성이 높은 선물 시장 환경 속에서 소음(Noise)과 신호(Signal)를 구분하고, 변화하는 시장 국면에 동적으로 대응하여 장기적으로 안정적인 누적 수익을 창출할 수 있는 최적의 매매 정책을 발견하는 것이다. 이를 통해 개발된 에이전트가 단순히 과거의 패턴을 모방하는 것을 넘어, 리스크를 관리하고 새로운 시장 상황에 적응하는 지능형 트레이딩 전략을 스스로 구축할 수 있는지 그 가능성을 탐색하고자 한다.

III. 핵심이론

A. PPO

PPO(Proximal Policy Optimization) 알고리즘은 TRPO(Trust Region Policy Optimization, 신뢰 영역 정책 최적화) 알고리즘의 복잡한 제약 최적화 문제를 피하고 정책 업데이트의 안정성을 높였다. TRPO 알고리즘은 정책 기반 강화학습 알고리즘의 핵심과제인 학습률에 따른 안정적인 학습을 제시했다. TRPO는 정책 성능을 대리하는 대리 목적 함수(surrogate objective function)을 최대화하되, 이전 정책과 새로운 정책 사이의 차이가 특정 범위 내 머물도록 강제한다. 이 차이는 KL divergence로 측정되며, 이 제약조건으로 신뢰영역을 정의할 수 있다. 신뢰 영역 내에서 정책을 업데이트하면, 새로운 정책의 성능이 이전 정책보다 나빠지지 않음을 이론적으로 보장된다. 하지만 이 알고리즘은 최적화에서 어려움이 있었기에, PPO는 TRPO의 이론적 안정적을 고수하되, 경량화된 최적화 기법을 적용해 활용성을 높였다.

- 대리 목적 함수와 확률 비율

정책 기반 알고리즘은 기대 보상을 나타내는 목적 함수 J_θ를 최적화한다. 하지만 J_θ를 직접 최적화하는 것은 어렵기 때문에, PPO와 TRPO는 대리 목적 함수 (surrogate objective function)를 최적화한다. 가장 기본적인 대리 목적 함수는 다음과 같이 정의된다.

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t \right] = \hat{\mathbb{E}}_t[r_t(\theta)\hat{A}_t]$$

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$$

A_t : 시간 t에서의 이점(advantage) 추정치

$r_t(\theta)$: 확률 비율(probability ratio)은 새로운 정책과 이전 정책 사이의 변화를 측정한다. 만약 $r_t(\theta) > 1$이면, 상태 s_t 에서 행동 a_t 를 선택할 확률이 새로운 정책에서 더 높다는 것을 의미한다. 반대로 $r_t(\theta) < 1$이면, 그 확률이 더 낮아졌음을 의미한다.

- 클리핑 오류 함수

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

PPO에서 가장 많이 사용되는 오류 함수는 확률 비율을 클리핑하는 방식이다. 기본 오류 함수인 $L^{CPI}(\theta)$는 A_t 가 양수일 때 $r_t(\theta)$가 무한정으로 늘어나는 경향이 있기 때문에 클리핑을 통해 억제한다. 이는 정책 성능에 대한 하한선을 형성해 정책 업데이트가 너무 멀리 변화할 때 이를 무시한다.

- GAE(Generalized Advantage Estimation)

PPO의 또다른 핵심은 이점 함수 $A(s,a) = Q(s,a) - V(s)$다. 이는 특정 상태 s에서 특정 행동 a를 취하는 것이 평균적인 행동보다 얼마나 더 좋은지를 측정한다. 단순한 누적 보상보다 상태가치가 베이스라인 역할을 해 경사도의 추정치의 분산을 크게 줄인다.

$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}$$

PPO에서는 $A(s,a)$를 추정하기 위해 GAE를 사용한다. GAE는 서로 다른 시간 길이의 이점 추정치들을 지수 가중 평균을 통해 조정한다. GAE는 두 개의 하이퍼파라미터 할인율 γ와 평활화 파라미터 λ에 의해 제어된다.

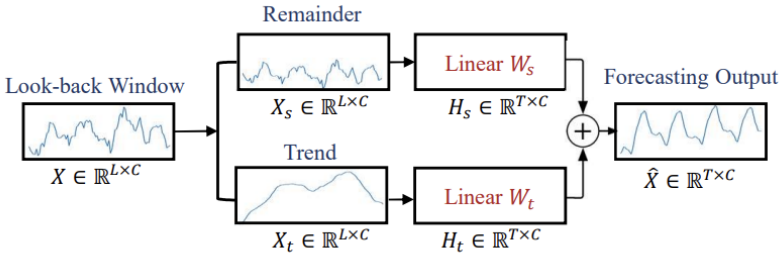
B. 신경망

본 프로젝트는 DLinear, CNN Transformer Hybrid Model, Informer을 시계열 예측에 사용한다.

1. DLinear

DLinear는 Transformer가 long-term time series forecasting(이하 long-term TSF) 테스트에 정말로 효과적인지에 대해 의문을 제기하며, 보다 시계열 데이터의 특성을 살려 long-term TSF에 효과적인 모델을 제안한다. Transformer 모델의 핵심적인 부분은 multi-head self-attention 메커니즘이다. 이는 길이가 긴 시퀀스 내의 요소 쌍 간의 의미적인 상관관계를 추출하는데 좋은 성능을 보인다. 하지만 이 과정은 '시간 순서에 관계없이' 작동한다. 따라서 연속적인 점들 사이의 순서 자체가 중요한 의미를 갖는 시계열 데이터의 시간 순서 정보를 무시할 수 있다.

DLinear는 시계열 분해(decomposition)과 단순한 one-layer linear network라는 단순한 구조를 갖는다. 이는 one-layer linear network가 미래 예측을 위해 과거 데이터를 통합하는 가장 간단한 형태의 네트워크라는 것과, 이전 연구들에 의하면 시계열 분해가 Transformer 계열 방법론을 사용한 TSF에 효과적이며 linear network를 포함한 다양한 모델의 성능을 올릴 수 있다는 점에서 착안한 구조이다.



(a) The whole structure of *DLinear*

[그림 3] DLinear 구조도

구체적인 과정은 다음과 같다. 우선 시계열 데이터를 Trend와 Remainder로 시계열 분해한다. 이렇게 분해한 각각의 데이터에 one-layer linear network를 통과시키고, 각 결과값을 더해 최종 예측 결과를 출력한다. 아래 수식에서 아래 첨자 t는 trend, s는 remainder를 의미한다. 데이터의 차원 L은 입력 시계열 데이터의 시퀀스 길이, C는 입력 채널 수, T는 예측 시퀀스 길이이다.

〈시계열 분해 (Decomposition)〉

$$X = X_t + X_s \in R^{L\times C}$$

〈one-layer linear network〉

$$H_t = W_t X_t \, , H_s = W_s X_s \qquad (\, W \in R^{T\times L} \, , H \in R^{T\times C} \,)$$

〈최종 예측 결과〉

$$\hat{X} = H_t + H_s \in R^{T\times C}$$

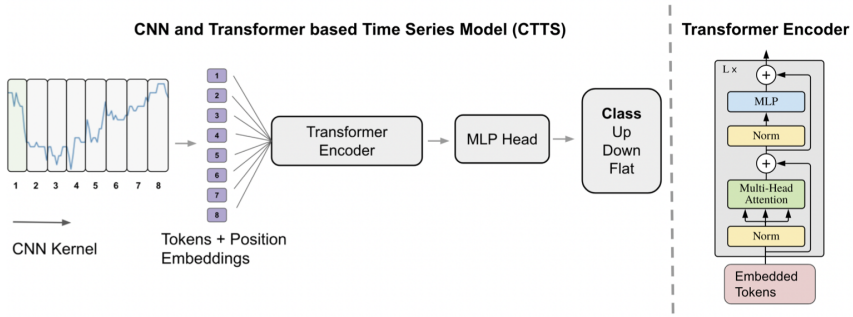
DLinear의 종류에는 신경망 가중치를 전체 데이터가 공유하는지 여부에 따라 DLinear-S와 DLinear-I 두 가지 종류가 있다. 만약 데이터의 변량(variate)이 각기 다른 특징을 갖는다면, 즉 각각 다른 경향성(trend)와 주기성(seasonality)를 갖는다면 같은 신경망 가중치를 공유하는 방법으로 좋은 성능을 낼 수 없기 때문이다.

DLinear-S: 모든 variate가 같은 Linear Layer를 공유한다.

DLinear-I: 각 variate가 각각의 Linear Layer를 가진다.

DLinear은 다음과 같은 장점이 있다. 가장 큰 장점은 모델 구조가 단순해서 모델이 가볍고 빠르다는 것이다.

2. CTTS(CNN + Transformer)



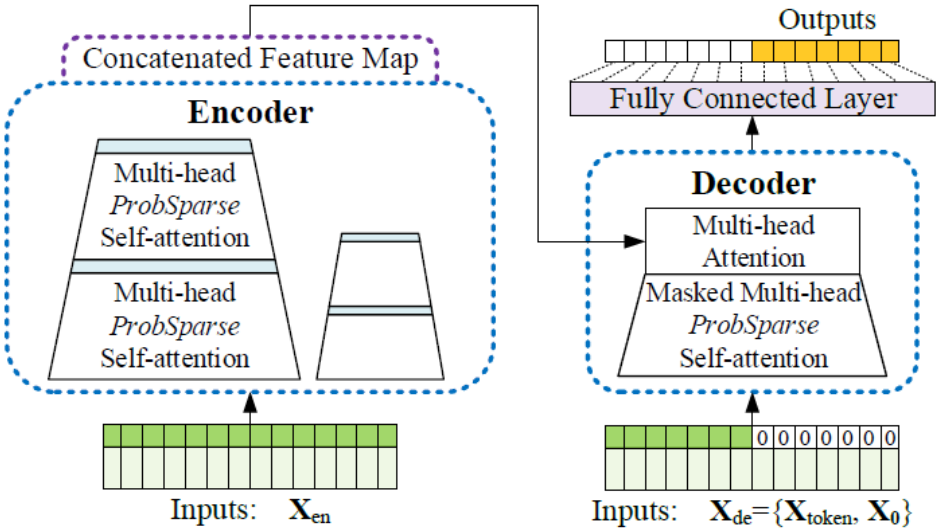
[그림 4] CTTS 모델 전체 구조도

시계열 데이터 분석의 핵심은 미세한 지역적인 패턴과 장기적인 패턴을 동시에 잡는 것이다. 일반적으로 딥러닝 분야에서 지역적인 패턴 추출은 합성곱 신경망 (CNN)이, 장기적인 패턴 추출은 트랜스포머(Transformer)가 사용된다. CNN은 합성곱 커널을 이용해 격자 형태의 행렬 데이터 셋에서 공간적인 정보 추출에 강력한 성능을 보인다. 이러한 특성은 패턴이 지역화되어 있는 시각적 데이터 처리에 효과적이다. Transformer 신경망은 Attention 메커니즘을 이용해 입력 시퀀스 전역에 걸쳐 있는 장기 종속성과 전역적인 정보 처리에 강점이 있다.

하지만 이 두 모델은 각각 명확한 한계를 가진다. CNN의 제한된 커넬 크기는 장기적인 추세를 놓치게 만들고, 트랜스포머의 전력적 관점은 미세한 지역의 변동성을 간과할 수 있다. CNN과 Transformer 모델을 합친 하이브리드 모델 CTTS(CNN and Transformer based time series modeling)는 합성곱 신경망이 데이터의 국소적인 특징을, 트랜스포머 신경망이 장기적인 특징을 잡아내, 서로의 한계를 보완하기를 기대한다.

CTTS 모델은 잔차 블록(residual block) 사이에 레이어 정규화를 적용하는 Post-LN 방식이 아닌, 잔차 블록 안에 레이어 정규화를 적용하는 Pre-LN 방식을 채택했다. 초기 트랜스포머에 사용된 Post-LN은 초기화 시점에 그래디언트가 불안정해 학습률 선택에 과하게 영향을 받는다. 이 문제를 해결하기 위해서 매우 작은 학습률부터 점진적으로 증가시키는 학습률 warm-up 단계가 적용되었다. Pre-LN은 Post-LN과 달리, 전반적인 그래디언트가 안정되어 학습률 웜업 단계없이도 원만한 학습을 가능하게 한다.

3. Informer



[그림 5] Informer 모델 구조도

시계열 예측 분야에서 RNN 기반 모델은 기울기 소실 문제로 인해 장기 의존성 학습에 한계가 있고, CNN 기반 모델은 고정된 필터 구조로 인해 시간에 따른 동적 변화를 충분히 반영하지 못한다. Transformer는 self-attention 메커니즘을 통해 장기 의존성 문제를 해결할 수 있는 잠재력을 보여주었지만, 긴 시퀀스 처리 시 연산 복잡도가 시퀀스 길이의 제곱에 비례하여 효율성 문제가 발생한다. 또한 다층 인코더 구조에서 메모리 병목 현상이 발생하고, 디코더의 순차적 예측 방식으로 인한 속도 저하와 누적 오차 문제도 존재한다.

Informer는 이러한 한계를 극복하기 위해 세 가지 핵심 기법을 도입했다. 첫 번째는 ProbSparse self-attention 메커니즘으로, "모든 query-key 쌍이 동일하게 중요하지 않다"는 통찰에서 출발한다. 각 query에 대해 attention 확률 분포가 균등분포와 얼마나 다른 지를 측정하는 중요도 지표를 개발하여, 상위 몇 개의 가장 중요한 query만을 선별해 attention을 계산한다. 이를 통해 정보 손실을 최소화하면서도 연산 복잡도를 $O(L^2)$에서 $O(LlogL)$로 대폭 감소시켰다.

두 번째는 self-attention distilling 기법으로, 메모리 병목 문제를 해결하기 위해 각 attention 레이어의 출력에 1차원 컨볼루션과 최대 풀링을 순차적으로 적용하여 시퀀스 길이를 절반으로 줄이면서도 핵심적인 시간적 패턴은 보존한다. 이러한 점진적 압축 과정을 통해 첫 번째 레이어에서는 전체 시퀀스를, 두 번째 레이어에서는 절반 길이의 압축된 정보를, 세 번째 레이어에서는 다시 절반으로 줄어든 정보를 처리하게 되어 전체 메모리 사용량을 거의 선형 수준으로 유지할 수 있다.

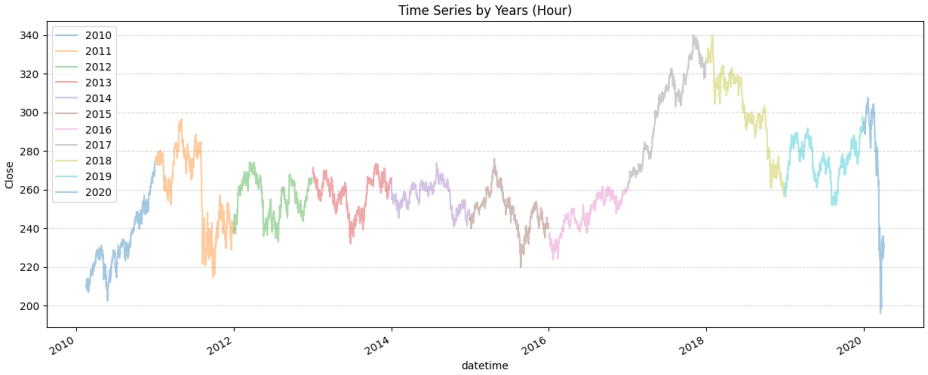
세 번째는 generative style decoder로, 기존 Transformer의 순차적 디코딩 방식 대신 시작 토큰과 예측 부분을 포함한 전체 시퀀스를 동시에 입력받아 한 번의 forward pass로 전체 출력을 생성한다. 이는 순차적 예측에서 발생할 수 있는 누적 오차를 방지하고 예측 시간을 시퀀스 길이에 비례하지 않고 거의 일

정하게 유지하여 추론 속도를 크게 향상시킨다. 실험 결과, Informer는 단변량 시계열 예측에서 기존 모델들을 상당한 차이로 앞섰으며, 특히 예측 길이가 길어질수록 성능 우위가 뚜렷하게 나타났다. 본 연구에서는 이러한 Informer의 장기 의존성 학습 능력과 효율성을 활용하여 강화학습 기반 트레이딩 시스템의 상태 표현 학습 성능을 향상시키고자 한다.

Ⅳ. 데이터 분석 및 전처리

A. 데이터 개요

본 연구에서 사용한 데이터는 코스피 200 미니 선물(KOSPI 200 Mini Futures)의 분봉 시계열 데이터로, 2010년 2월 16일부터 2020년 4월 3일까지 약 10년간의 거래 데이터를 포괄한다. 해당 데이터는 한국거래소 정규 거래시간 동안의 1분 단위 가격 정보를 담고 있으며, 시가(Open), 고가(High), 저가(Low), 종가(Close), 거래량(Volume)의 표준 OHLCV 형태로 구성되어 있다. 특히 2020년도 데이터는 COVID-19 팬데믹 발생 시점의 극심한 시장 변동성을 포함하고 있어, 위기 상황에서의 선물 시장 동향을 분석할 수 있는 중요한 자료로 활용된다.



[그림 6] 코스피 200 미니 선물 월별 시간별 종가 추이 (2010-2020)

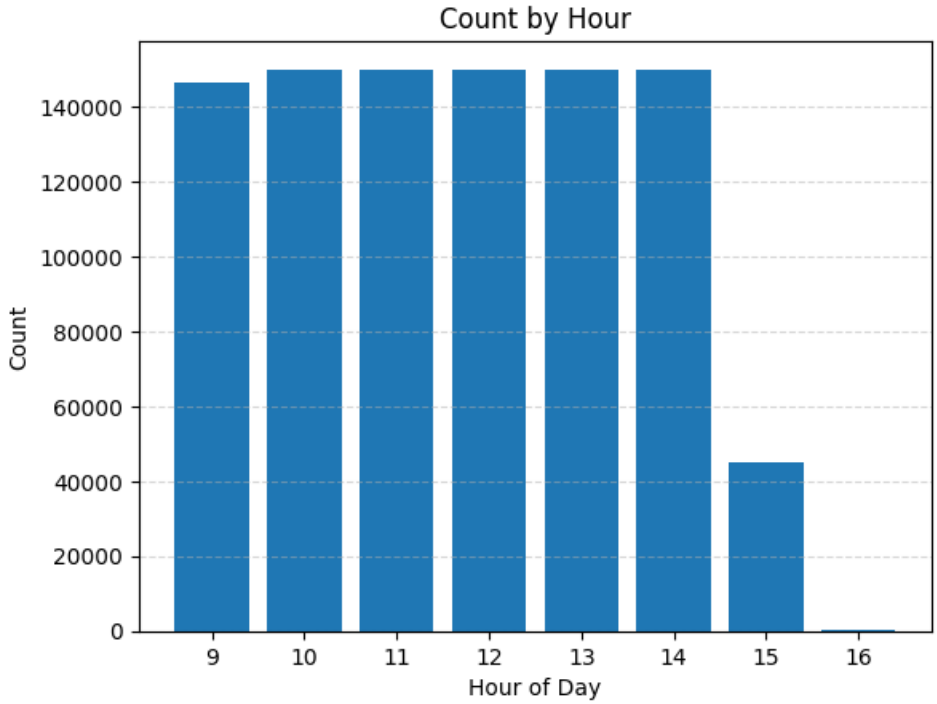
B. 시장 추세 분석

2010년부터 2020년까지의 코스피 200 미니 선물 월별 종가 시계열을 분석한 결과, 각 시기별로 서로 다른 특성을 가진 시장 국면들이 연속적으로 나타났다. 2010-2011년 초기 회복기에는 금융위기의 여파에서 벗어나며 점진적인 상승세가 관찰되었고, 2012-2015년 안정기에는 큰 변동 없이 완만한 등락을 반복하는 횡보장이 지속되었다.

2016-2018년 들어서는 국내외 경제 회복과 유동성 공급 확대의 영향으로 뚜렷한 상승 모멘텀이 나타났으며, 이는 전체 분석 구간에서 가장 강력한 불장을 형성했다. 하지만 2019년부터는 미중 무역갈등과 글로벌 불확실성이 커지면서 시장 분위기가 바뀌기 시작했고, 2020년 초 코로나19 팬데믹 충격으로 인해 급격한 폭락과 극심한 변동성을 경험한 후 빠른 반등이 이어졌다.

각 연도별로 구분된 시계열 패턴에서도 시기마다 변동 폭과 방향성이 뚜렷하게 달라지는 것을 확인할 수 있으며, 특히 2020년 3월을 기점으로 한 급변하는 시장 상황이 두드러진다. 이처럼 안정적인 구간부터 극한 상황까지 포함하는 다양한 시장 환경은 강화학습 모델이 일반적인 거래 상황뿐만 아니라 위기 대응 능력까지 함께 학습할 수 있는 이상적인 데이터셋을 제공한다.

C. 데이터 특성



[그림 7] 전체 데이터의 시간별 빈도

시간대별 데이터 분포 분석을 통해 거래시간 변경의 영향을 확인할 수 있었다. 막대 그래프를 확인한 결과, 15시 데이터의 경우 상대적으로 데이터 수가 부족한 것으로 나타났는데, 이는 장 마감 시간이 정시가 아닌 15시 45분(2016년 8월 이전) 또는 15시 30분(2016년 8월 이후)이기 때문이다. 16시 데이터의 경우에는 주로 수능으로 인한 장 개폐 지연으로 발생하였으며, 특히 2017년에는 포항 지진으로 인해 수능이 연기되면서 16시까지 장이 열린 날이 2회 기록되었다.

분석 기간 중 한국거래소의 거래시간 변경이 두 차례 발생하였다. 2016년 8월 1일부터 장 마감 시간이 기존 15:15에서 15:45로 30분 연장되었고, 이후 현재의 15:30으로 조정되었다. 이러한 제도적 변화는 일일 거래 데이터의 길이에 직접적인 영향을 미치므로 시계열 모델링 시 중요한 고려사항이다. 9시 데이터의 경우에는 매년 대학수학능력시험일과 신정(1월 2일) 연초에 장 시작이 1시간 지연되면서 해당 시간대 데이터가 누락되는 패턴을 보인다.

D. 데이터 전처리

데이터 전처리 과정에서는 먼저 장 마감 직전 시스템 지연으로 발생하는 15:06분 및 16:06분 데이터를 각각 15:05분 및 16:05분 데이터로 통합 처리하

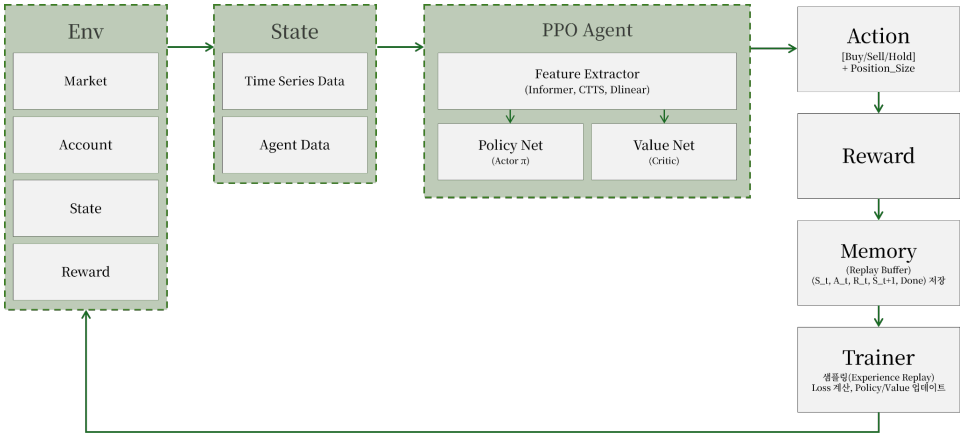
였다. 결측치 처리의 경우, 전체 타임스탬프를 연결하여 누락된 시점을 식별한 후 NaN 값으로 패딩을 실시하되, 서킷 브레이커 발생 시점을 포함하면서도 장 마감 직전 10분간 데이터는 공백을 보완하는 방식을 채택하였다.

특이 데이터로는 2010년 7월 16일 데이터가 있는데, 해당 일자는 15:15 데이터만 존재하며 일반적인 패턴과 달리 OHLC 값이 모두 상이하고 익일 데이터의 전일종가와 불일치하는 이상 현상을 보여 분석에서 제외하였다. 코로나19 사태로 인한 2020년 3월 13일과 19일의 서킷 브레이커 발생 데이터는 시장의 비정상적 상황을 반영하므로, 해당 기간의 29개 결측치를 보간법으로 처리하는 것보다는 제외하는 것이 논리적으로 적절하다고 판단하였다.

시간 기준으로 타임스텝 당 1-2개 수준의 소규모 결측치에 대해서는 "마지막 가격이 유지된다"는 시장 원리에 따라 전진 채움법(Forward Fill)을 적용하였다. 이는 급격한 가격 변동보다는 보수적 접근을 통해 모델의 안정성을 확보하기 위한 전략이다. 전처리 완료 후에는 타임스탬프 연속성, OHLCV 데이터의 논리적 일관성, 이상치 탐지, 거래량과 가격 데이터 간 상관관계 등을 종합적으로 검증하여 강화학습 모델 학습에 적합한 고품질 데이터셋을 구축하였다.

Ⅴ. 구현

A. 구조도



[그림 8] 전체 프로젝트 구조도

B. Agent 구현

PPO 알고리즘을 기반으로 구현했다.

행동 공간 정의 및 제약 조건

본 연구의 에이전트는 선물 시장에서 능동적으로 진입 및 청산 시점을 관리할 수 있도록 설계되었다. 이를 위해 행동 공간(Action Space)은 -10(10계약 매도)부터 +10(10계약 매수)까지의 정수 값을 갖는 이산 공간(Discrete Space)으로 정의하였으며, 0은 포지션을 유지하는 행동(Hold)을 의미한다.

그러나 실제 투자 환경의 현실성을 반영하기 위해 다음과 같은 제약 조건을 설정하였다. 초기 자본금 30,000,000원을 기준으로, 에이전트가 보유할 수 있는 최대 포지션의 상한을 10계약으로 제한하였다. 이 제약 조건으로 인해, 에이전트의 현재 포지션 상태에 따라 행동 공간 내의 일부 행동이 일시적으로 허용되지 않는 행동(Invalid Action)이 될 수 있다. 예를 들어, 에이전트가 이미 매수 7계약 포지션을 보유한 상태에서 추가로 4계약을 매수하는 행동(+4)은 총 보유 한도인 10계약을 초과하므로 유효하지 않다.

유효하지 않은 행동 처리 기법: 액션 마스킹

강화학습에서 이와 같이 환경의 제약 조건으로 발생하는 유효하지 않은 행동을 처리하기 위해 일반적으로 3가지 접근법이 사용된다.

1. **행동 무시 (Ignoring)**: 유효하지 않은 행동이 선택되면 이를 무시하고 이전 상태를 그대로 유지한다.
2. **페널티 부여 (Penalizing)**: 유효하지 않은 행동 선택 시, 에피소드를 즉시 종료(done=True)하고 큰 음수 보상(페널티)을 부여하여 에이전트가

해당 행동을 회피하도록 학습시킨다.

3. **액션 마스킹 (Action Masking)**: 정책이 행동을 선택하기 전 단계에서, 유효하지 않은 행동들의 확률 값을 원천적으로 0으로 만들어 선택 가능성을 배제한다.

이론적인 관점에서 페널티 방식은 에이전트가 시행착오를 통해 환경의 제약 조건을 스스로 학습하게 한다는 점에서 타당성을 갖는다. 그러나 이는 학습의 주 목표인 수익 극대화 전략 탐색을 저해하고, 불필요한 탐험으로 인해 학습 속도를 현저히 저하시키는 비효율을 야기할 수 있다.

최근 연구에 따르면, PPO(Proximal Policy Optimization)와 같은 정책 기반 알고리즘에서 액션 마스킹 방식이 페널티 방식보다 더 빠른 초기 수렴 속도와 높은 성능을 보이는 것으로 보고되었다. 마스킹은 에이전트가 명백히 불가능한 행동을 탐험하는 데 시간을 낭비하지 않도록 하여 탐험 효율성(Exploration Efficiency)을 극대화하고, 결과적으로 더 빠르고 안정적인 정책 학습을 가능하게 한다.

따라서 본 연구에서는 학습 효율성을 최우선으로 고려하여, 유효하지 않은 행동을 처리하는 기법으로 액션 마스킹을 채택하였다. 이를 통해 에이전트는 주어진 포지션 한도 내에서 유효한 행동만을 대상으로 최적의 전략을 탐색하게 된다.

C. 환경 구현

환경에서 투자자의 계좌와 선물 시장의 흐름 제어를 나누어 구현했다.

선물 시장 제어 클래스는 강화학습의 메인 환경이다. 현재 상태에서 행동을 받아 다음 상태, 보상, 종료 여부 계산해 에피소드를 진행하는 역할을 하도록 구현하였다. 데이터를 제공하는 역할을 하므로, 학습에 사용할 데이터와 학습에 사용할 기간 및 환경 관련 초기화 값을 모두 전달받고, 데이터를 기간에 따라 잘라서 학습에 사용할 수 있는 데이터셋으로 만들어 State를 생성한다. 해당 클래스는 기본적인 선물 계약 단위, 만기일, 거래 비용, 슬리피지, 시장 상태와 같이 주로 선물 시장과 관련된 변수들을 가진다. 실시간으로 현재 스텝과 현재 스텝의 선물 가격, 만기일까지 남은 날짜, 현재 상태를 나타내는 info 등으로 현재 환경을 추적한다. 이외에 보상 관련 함수와 페널티, 종료 여부를 계산하는 함수, 에이전트의 성과를 평가하는 변수들을 통해 에피소드를 진행하고 에이전트의 성과를 평가할 수 있다. 시장 환경의 가장 중요한 역할은 강화학습의 환경으로써 step을 정확하게 진행하는 것이고, 내부적으로는 done 확인 및 현재 상태 제공, 에이전트의 행동 제약이 핵심적인 구현부이다. 구체적인 동작 과정은 다음과 같다.

매 step마다 다음 데이터와 에이전트의 행동에 따라 시장 상태와 계좌를 업데이트하고, 종료 조건 및 강제 청산 여부 계산, 거래 결과 저장을 하며, 에이전

트의 성과를 추적한다. 이후 보상을 계산하고, 다음 state를 생성한다. 이때 다음 스텝에서의 행동 제약을 계산해 다음 스텝에서 적절한 행동을 선택하도록 한다. 이때 에이전트의 행동을 하드코딩으로 제한하여 보다 안정적인 학습을 할 수 있도록 하였다. 에이전트가 보유할 수 있는 계약 수에 제한이 존재하고, 현재 에이전트가 보유 중인 자금에 따라서 불가능한 행동이 있으므로 각 스텝마다 에이전트가 선택할 수 있는 행동에 제약을 주고, 리스크 제한 초과 시 강제 청산을 하도록 하여 과도한 손실을 강제로 방지하였다.

에피소드 종료 조건은 아래와 같이 여러 가지 가능한 상황을 추적해 info 변수에 저장하였다. 구체적인 종료 조건은 아래 E-에피소드 종료 및 청산 설계 부분에 기술하였다. 이때 만기일은 매월 만기일 정의에 따라 전체 입력 날짜에 존재하는 모든 만기일을 한 번에 계산하는 함수를 사용하였다. 미니 코스피200 선물의 만기일은 매월 두 번째 목요일로 정해져 있고, 만약 해당 날짜가 거래일이 아니라면 직전 거래일이 만기일이 된다.

계좌 클래스는 선물 거래를 하는 투자자의 계좌, 포지션, 손익을 관리하고, 새로운 계약 체결 및 계약 청산 기능을 하도록 구현하였다. 초기 자산과 계약 포지션, 증거금 비율, 거래 비용 등의 정보를 가지고 있고, 매 스텝마다 현재 계좌와 포지션, 보유 중인 계약에 대한 가치 및 실현 손익과 미실현 손익 등을 업데이트한다.

계약 체결과 계약 청산 기능은 실제 선물 거래 계좌창을 참고하여 비슷하게 작동하도록 구현하였다. 신규 계약 진입 시 위탁 증거금이 가용 계좌에서 위탁 증거금 계좌로 이동하며, 청산 시 위탁 증거금이 위탁 증거금 계좌에서 가용 계좌로 이동한다. 미결제약정 리스트로 보유 계약을 관리하며, 청산 시 선입선출 방식으로 먼저 체결된 계약부터 청산되도록 한다. 청산 시 발생하는 실현 손익은 가용 계좌에 반영되고 따로 저장한다. 보유 중인 미결제약정에 대한 유지 증거금과 아직 실현되지 않은 미실현 손익은 매 스텝마다 업데이트된다. 모든 실현 손익은 거래 비용과 슬리피지를 반영한 순 실현 손익으로 계산한다. 하루 치 장이 종료되면 실제 선물 거래와 같이 일일정산이 이루어지고, 그에 따라 계좌, 실현 손익, 미실현 손익을 업데이트한다.

본 연구에서 에이전트가 학습할 매매 전략의 근간은 **스윙 트레이딩(Swing Trading)**이다. 스윙 트레이딩은 단기적인 가격 변동이 형성하는 추세의 전환점, 즉 '스윙(Swing)'을 포착하여 수익을 창출하는 중단기 투자 방법론이다. 이 전략은 기업의 내재가치보다 시장의 심리, 모멘텀, 추세의 강도를 계량화한 기술적 분석 지표를 핵심적인 판단 근거로 삼는다.

이러한 스윙 트레이딩의 의사결정 과정을 강화학습 에이전트를 통해 자동화하고 최적화하기 위해, 에이전트의 학습 환경을 다음과 같이 설계하였다. 첫째, 스윙 트레이더가 주로 활용하는 이동평균선, RSI, MACD 등 핵심 기술 지표들을 에이전트가 시장을 인식하고 판단을 내리는 다차원 상태(State) 정보로 정의하였다. 둘째, 며칠에서 수 주에 걸친 중단기적 매매 호흡을 모사할 수 있도록

에피소드의 길이를 설정하고, 이에 부합하는 보상 함수를 적용하여 전략의 목표를 명확히 하였다.

D. State 설계

실제 강화학습 상태를 처리하는 State 클래스를 구현해 에이전트에게 제공하는 상태를 적절하게 제공할 수 있도록 하였다.

학습에 사용하는 state는 시계열 데이터셋과 에이전트 정보가 결합된 상태이다. 시장 정보를 가진 시계열 데이터셋과 함께, 에이전트가 현재 잔고와 포지션 등 자신의 상태를 고려해 행동을 선택하고 자산 관리를 학습하도록 한다는 목적으로 state를 이렇게 두 가지 부분으로 설계하였다.

시계열 선물 시장 데이터셋이 선물 가격 데이터에 추가로 여러가지 경제적 지표를 계산한 변수들이 포함되어있는 데이터 셋이므로, State 클래스에서는 이 중 어떤 변수를 학습에 사용할 것인지 입력 받아 해당 변수만 추출해 시계열 데이터셋을 구성한다. 시계열 변수는 로버스트 스케일링을 사용해 정규화한다. 에이전트 정보는 현재 잔고, 포지션, 계약 체결 강도, 거래 비용 등이 있다.

학습에 사용한 시계열 선물 시장 변수와 에이전트 변수는 다음과 같다. 시계열 변수의 타임스텝은 1분 간격이다.

[표 2] 시계열 선물 시장 상태 변수

변수명	설명
close	타임스텝(1분 간격) 마다의 종가
high	타임스텝 동안의 최고가
low	타임스텝 동안의 최저가
volume_change	직전 타임스텝 대비 거래량 변화량
ema_5	최근 5개 타임스텝의 종가를 기반으로 계산한 지수이동평균선(Exponential Moving Average)
ema_20	최근 20개 타임스텝의 종가를 기반으로 계산한 지수이동평균선
ema_cross	단기 지수이동평균(ema_5)과 장기 지수이동평균(ema_20)의 차이 (ema_5−ema_20). 이 지표의 부호와 크기를 통해 추세의 방향과 강도를 파악할 수 있다. - 골든 크로스: ema_cross > 0, 강세장 신호 - 데드 크로스: ema_cross < 0, 약세장 신호
rsi	상대 강도 지수(Relative Strength Index). 일정 기간 동안 주가의 상승 압력과 하락 압력 간의 상대적인 강도를 나타낸다. 최근 14개 타임스텝을 기준으로 사용하였다. 주로 과매도와 과매수 상태를 파악하는데 사용하는 지표이다.
%K, %D	스토캐스틱 오실레이터(Stochastic Oscillator). 일정 기간 동안의 주가 변동 범위(최고가와 최저가) 내에서 현재 종가가 어느 위치에 있는지를 백분율로 나타내는 모멘텀 지표. 과

	매도와 과매수 상태를 파악하는데 사용하는 지표이다. - %K: 현재 가격이 최근 14개 타임스텝의 가격 범위 중 어디에 위치하는지를 나타낸다. - %D: %K 값을 이동평균하여 계산한 값으로, %K보다 부드럽게 움직이며 시장의 단기적인 노이즈를 줄여준다. 이동평균 윈도우 사이즈는 3을 사용하였다.
cci	상품 채널 지수(Commodity Channel Index). 주가가 이동평균선에서 얼마나 떨어져 있는지를 측정하여 추세의 방향과 강도를 파악하는 모멘텀 오실레이터이다.
atr	평균 실제 범위(Average True Range). 주가의 변동성을 측정하는 지표이다. 가격의 방향성은 알려주지 않으며, 가격이 얼마나 활발하게 움직이는지를 나타낸다. 시장 위험도를 측정하는 지표이다.
bb_width	볼린저 밴드 폭(Bollinger Band Width). 볼린저 밴드의 상단 밴드와 하단 밴드 사이의 폭을 나타내는 지표이다. 이 폭은 중간 밴드(이동평균선)를 기준으로 정규화된다. 가격 변동성을 측정하는 지표이다.
obv	온밸런스 거래량(On-Balance Volume). 주가가 상승한 날의 거래량은 더하고, 하락한 날의 거래량은 빼서 누적시킨 값이다. '거래량은 주가에 선행한다'는 전제를 바탕으로 하며, 자금의 유입과 유출을 통해 주가 추세를 예측하는 데 사용하는 지표이다.

[표 3] 에이전트 상태 변수

변수명	설명
current_position	현재 에이전트의 포지션 (long: +1, short: -1, 보유 계약 없음: 0)
excution_strength	계약 체결 강도 (보유 계약 수)
n_days_before_ma	만기일까지 남은 일 수
realized_pnl	현재 타임스텝에서의 실현 손익 (pt)
unrealized_pnl	현재 타임스텝에서의 미실현 손익 (pt)
available_balance	현재 타임스텝에서의 가용 잔고 (pt)
cost_ratio	초기 자산(30,000,000원) 대비 총 수수료의 비율 (%)

E. Reward, Done 설계

보상 설계

최종 보상 함수 R_t는 수익 (R_{profit}), 위험 (R_{risk}), 후회 (R_{regret}) 세 가지 요소의 가중합과 상황에 따른 페널티로 구성했다.

1. 수익 구성 요소

$$R_{profit,t} = log(P_{N,t}) + log(P_{U,t} - P_{U,t-1})$$

$P_{N,t}$: t 시점의 실현 손익(realized P&L)

$P_{U,t} - P_{U,t-1}$: t 시점의 미실현 손익의 변화량 (unrealized P&L)

log : $sgn(x) \cdot ln(1+|x|)$, 수치 안정성을 위해 적용된 로그 함수

2. 위험 구성 요소

$$R_{risk,t} = D_t(r_t) \quad s.t. \; r_t = log(P_{PV,t}) - log(P_{PV,t-1})$$

D_t : 위험 조정 수익률 변화 함수 (Differential Sharpe Ratio, DSR)

$P_{PV,t-1}$: 보유 자산 + 미실현 손익으로 구성된 포트폴리오 가치의 변화량

- Differential Sharpe Ratio (DSR)

Sharpe Ratio는 투자 성과를 판단하는 대표적인 지표다. 하지만 이 지표는 전체 에피소드의 평균 수익률과 표준편차를 이용하기 때문에 즉각적인 피드백이 필요한 강화학습 상황에는 적합하지 않다. DSR은 이런 한계를 극복하기 위해 Moody & Saffell (2001)에 의해 고안된 지표로, 샤프 지수의 미분값이다. DSR은 현재의 행동으로 인해 다음 스텝의 샤프 지수가 얼마나 변할 것인가를 측정한다.

$$A_t = (1-\eta) * A_{t-1} + \eta * r_t$$
$$B_t = (1-\eta) * B_{t-1} + \eta * r_t^2$$
$$\eta = 2/(span+1)$$

A_t : 1차 미분값, 평균

B_t : 2차 미분값, 분산

η : EMA 업데이트를 위해 사용되는 파라미터

$$D_t(r_t) = \frac{(B_{t-1} - A_{t-1}^2)(r_t - A_{t-1}) - \frac{1}{2}A_{t-1}(r_t^2 - B_{t-1})}{(B_{t-1} - A_{t-1}^2)^{3/2}}$$

DSR 업데이트 식

3. 후회 구성 요소

$$R_{regret,t} = \begin{cases} log(|\Delta p_t|) & \text{if } Pos_{t-1} = 0 \text{ and } Pos_t = 0 \\ 0 & \text{otherwise} \end{cases}$$

ΔP_t : 시장의 가격 변화

B_t : 2차 미분값, 분산

η : EMA 업데이트를 위해 사용되는 파라미터

포지션을 보유하지 않았을 때 시장이 움직인 것에 대한 기회비용(후회)을 나타낸다.

4. 보너스 지표

선물 시장의 이벤트에 따른 패널티, 스윙의 목표 달성 (5% 수익률)에 따른 보너스를 추가했다.

[표 4] 보너스 표

마진콜	유지 증거금이 부족해진 상황 파산이 발동되기 때문에 거의 발동되지 않지만, 방지용으로 추가	−2.0
파산	가용 자산이 0인 상황	−5.0
만기일	만기일 마지막 틱까지 포지션을 보유하고 있는 상황	−0.5
목표 달성	수익률 5%를 넘은 상황	+2.0

5. 최종 보상

$$R_{\text{base},t} = w_p * R_{\text{profit},t} + w_r * R_{\text{risk},t} - w_g * R_{\text{regret},t}$$

$$R_t = R_{\text{base},t} + \begin{cases} P_{\text{bankrupt}} & \text{if 파산} \\ P_{\text{margin call}} & \text{if 마진콜} \\ P_{\text{maturity}} & \text{if 만기일 강제청산} \\ B_{\text{goal}} & \text{if 목표수익 달성} \\ 0 & \text{otherwise} \end{cases}$$

w_p : 수익의 가중치 (=0.4)

w_r : 리스크의 가중치 (=0.4)

w_p : EMA 업데이트를 위해 사용되는 파라미터 (=0.2)

기본 보상에 이벤트 보너스를 더한 최종 보상 함수는 다음과 같다.

에피소드 종료 및 청산 설계

적절한 에피소드 길이 설정을 위하여, 현실 시간에서 2주 가량되는 3,000 스텝을 에피소드 최대 길이로 설정했다.

선물 시장에 존재하는 만기일, 당일 손익 전환 같은 다양한 이벤트에 따른 청산 및 강제 종료 조건들은 아래 표와 같다.

[표 5] 에피소드 상황 별 종료, 청산 기준 표

상황	강제 종료	청산	특징
마진콜	True	True	
파산	True	True	
만기일	True	True	
목표 달성	False	False	강제 종료하고자 했지만, 너무 짧은 에피소드 길이 때문인지 학습이 제대로 이루어지지 않았다.
당일 마지막 틱	False	False	미실현 손익을 실현 손익으로 전환
최대 길이	True	True	

F. 신경망 설계

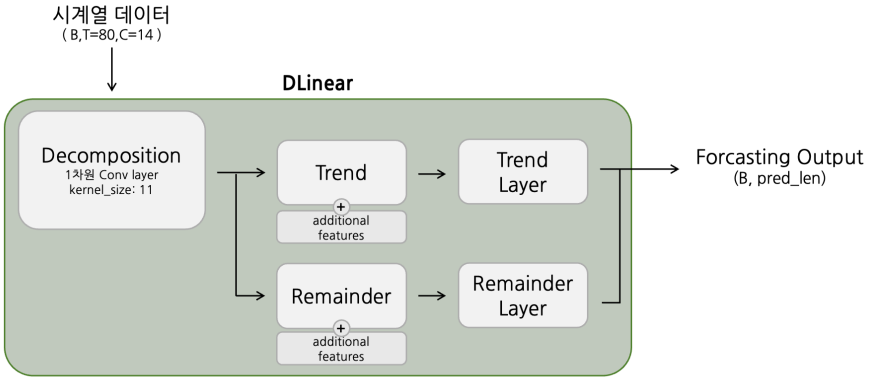
1. DLinear

시계열 선물 시장을 다루는 신경망으로 DLinear를 사용한다. 학습에 활용하는 요인들을 모두 입력하지만, 시계열 데이터 예측에 특화된 DLinear의 특성을 고려해 'close'가격만 예측하고 나머지 feature들은 예측에 참고하는 정보로 이용한다.

시계열 분해 부분(Decompose) 에서는 예측 목표인 'close'가격에 대해서만 시계열 분해를 한다. 나머지 feature 중에서 완전한 시계열 데이터라고 할 수 없는 feature에 대해서는 시계열 분해를 하면 안되고, 최종 목표가 close 가격의 시계열 예측값이기 때문이다. 슬라이딩 윈도우 방식으로 시계열 데이터의 Trend를 분리하고, 원본 데이터에서 Trend를 뺀 것이 Remainder이다. 입력 데이터가 torch.tensor 인 것을 고려해 1차원 합성곱 레이어 nn.Conv1d 로 슬라이딩 윈도우를 구현한다. 이때 시계열 데이터의 맨 앞과 맨 뒷 부분의 시계열 분해가 정상적으로 이루어지도록 kernel size에 맞춰 양 끝단 값으로 padding한다. 기본적인 시계열 분해를 위해 컨볼루션 레이어의 가중치를 모두 같은 값으

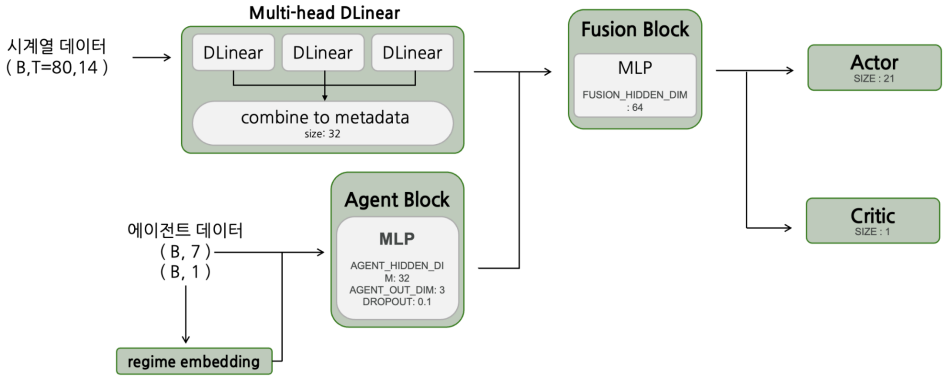
로 지정하고 학습할 수 없도록 설정한다. 시계열 분해를 하는 가중치도 학습할 수 있게 하여 시계열 분해도 최적화하도록 할 수 있다. 이때는 위의 가중치 초기화와 학습 불가능 설정 과정을 생략한다.

단일 DLinear 모델은 입력 시퀀스 길이와 예측 길이를 입력 받고 Trend와 Remainer 각각의 one-layer linear network를 갖는다. 입력 받은 시계열 데이터의 첫 번째 channel이 예측하는 feature이다. 해당 feature를 위의 Decomposition 레이어에 통과시켜 시계열 분해하고, 분해한 결과에 다른 feature를 결합한 Trend와 Remainder 데이터를 각각의 linear layer에 통과시킨다. 이렇게 얻은 예측 Trend와 Remainder를 더해 최종 예측 값을 출력한다.



[그림 9] 단일 DLinear 구조도

기본적인 단일 DLinear 모델에 Multi-head 모델 아이디어를 결합해 더 안정적인 예측값을 얻는 것을 목표로 단일 DLinear 모델 여러 개의 예측값으로 새로운 결과를 출력하는 모델을 고안하였다. 동일한 입력 데이터에 대해 여러 개의 DLinear 모델 각각이 하나의 head가 되어 서로 다른 예측 길이로 예측값을 출력하고, 모든 head의 결과를 종합해 최종 결과값을 얻는다. 단순히 예측값을 평균내는 방법과, 예측값에 대해 기초 통계량과 경제적 지표를 계산해 새로운 결과를 출력하는 방법 총 두 가지로 구현하였다. 단순 평균 방법은 서로 다른 예측 길이를 가진 예측값의 평균을 내어 더 안정적인 예측값을 얻을 수 있다. 새로운 지표(meta data)를 계산하는 방법은 기본적인 통계량과 경제적 지표에 더해, 단기 예측 값과 장기 예측 값의 특징을 살린 경제적 지표를 학습에 이용할 수 있도록 한다. 강화학습 에이전트가 단순 시계열 데이터가 아닌 경제적 지표를 이용해 학습하도록 해 더 좋은 성능을 기대하는 목적으로 구현하였다.



[그림 10] Multi-head DLinear 기반 Actor-Critic Network 구조도

Multi-head DLinear에서 여러 길이의 예측 값을 이용해 구하는 meta feature는 다음과 같다.

[표 6] Multi-head DLinear의 예측 통계량

변수명	설명
mean	예측 시퀀스 전체의 평균
std_dev	예측 시퀀스 전체의 표준편차. 변동성 지표이다.
total_return	예측 시퀀스의 절대적인 수익량 (예측값의 마지막 값 - 첫 번째 값)
total_return_rate	총 수익률 (total_return / 첫 번째 값)
max_drawdown	예측 기간 중 최고점에서 가장 많이 하락한 비율
final_pos_in_range	예측기간의 최저점과 최고점을 각각 0, 1이라고 했을 때, 마지막 값의 위치
linreg_slope	예측 시퀀스에 대한 선형 회귀선의 기울기
pred_trend_quality_ratio	예측값 변화량의 평균을 표준편차로 나눈 값. 추세의 일관성 또는 안정성을 나타낸다.

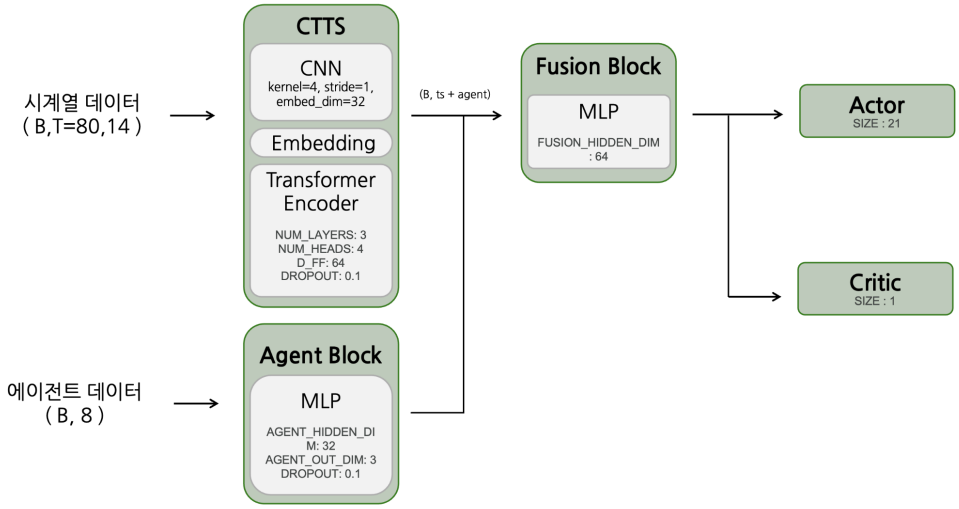
예측값을 통해 구한 위 표의 8개의 통계량에 대해 아래 4가지 관점으로 meta feature를 생성해, 총 $8 \times 4 = 32$개의 meta feature를 만든다.

- **평균**: 해당 지표의 전반적인 기대치
- **표준편차**: 해당 지표가 예측 길이에 따라 얼마나 변동하는지
- **단기 값**: 가장 짧은 예측 길이에서의 지표 값. 단기적인 특성을 나타낸다.
- **장기 값**: 가장 긴 예측 길이에서의 지표 값. 장기적인 특성을 나타낸다.

2. CTTS(CNN+Transformer)

State를 학습하기 위해서는 2 가지 신경망이 필요하다. 시계열 선물 시장의 흐름을 읽는 신경망으로 CTTS를, 나머지 에이전트의 잔고 상황을 읽는 신경망

으로 MLP를 사용했다. 각기 다른 신경망에서 나온 context vector를 가볍게 퓨전해 정책과 가치를 얻는다. 전체 구조는 아래와 같다.

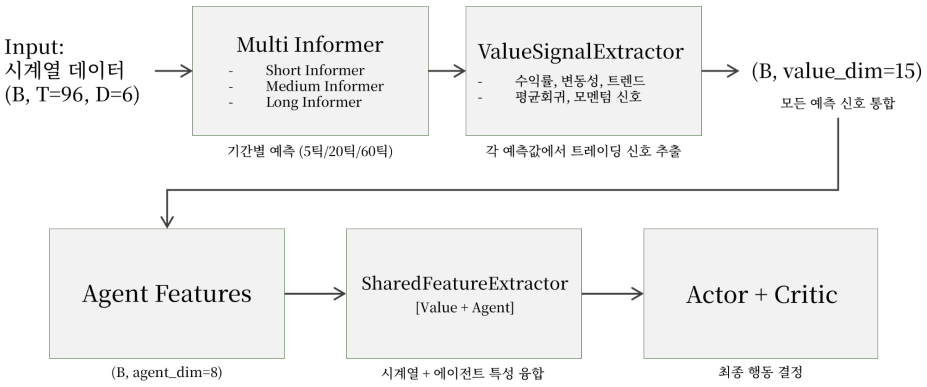


[그림 11] CTTS 기반 Actor-Critic Network 구조도

CTTS 기반 액터 크리틱 신경망은 다음과 같은 효과를 기대한다. 80 틱의 선물 시장의 시장 데이터와 기술적 분석 지표를 이용해 미니 선물 시장의 흐름을 읽어낸다. 에이전트의 단기적인 시점 지표, 자산 상황, 포지션 상황을 통해 수익을 위해 필요한 다음 행동 정보를 추출한다. 이 두 지표를 합쳐 현재 시장 상황에서의 정책과 현재 상태의 가치를 만들어 트레이딩에 사용된다.

3. Informer

본 연구에서는 Informer의 장기 시계열 예측 능력을 강화학습 트레이딩에 활용하기 위해 MultiInformer 구조를 설계했다. 전체 시스템 구조는 다음과 같다.



[그림 12] Informer 기반 Actor-Critic Network 구조도

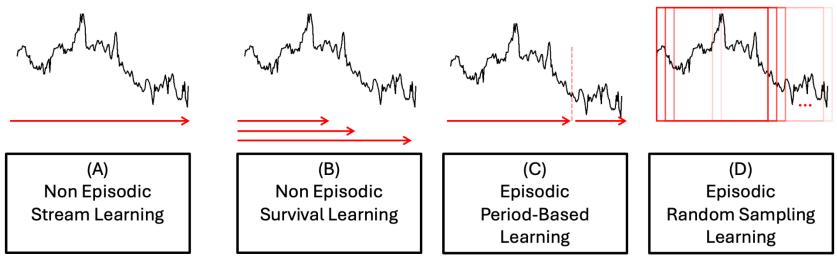
MultiInformer는 서로 다른 예측 기간(5틱, 20틱, 60틱)을 가진 세 개의 Informer 모델을 병렬로 운용하여 단기적 시장 노이즈부터 장기적 트렌드까지 다양한 시간 척도의 정보를 포착한다. 모든 Informer는 동일한 96개 시점의 과거 데이터를 입력으로 받지만, 각기 다른 out_len 파라미터를 통해 서로 다른 예측 기간에 특화되도록 학습된다.

각 Informer는 d_model=512, n_heads=8, e_layers=2의 공통 파라미터를 사용하여 충분한 표현력을 확보하면서도 계산 효율성을 유지했다. 핵심적으로 ProbSparse self-attention 메커니즘을 통해 중요한 query만을 선별하여 attention을 계산함으로써 긴 입력 시퀀스(96 시점)를 효율적으로 처리하면서도 장기 의존성 정보를 보존할 수 있다. 또한 각 인코더 레이어에서 self-attention distilling을 적용하여 1차원 컨볼루션과 최대 풀링을 통해 시퀀스 길이를 점진적으로 압축한다. 이를 통해 메모리 사용량을 대폭 감소시키면서도 핵심적인 시간적 패턴은 유지한다.

시간 정보의 효과적인 활용을 위해 별도의 TimeFeatureGenerator 클래스를 구현하여 Informer의 시간 인코딩을 처리한다. 이 모듈은 실제 날짜 데이터로부터 정규화된 시간 특성을 생성하며, 일별 데이터의 경우 3차원의 시간 특성을 자동으로 생성한다. 각 Informer의 예측 결과는 ValueSignalExtractor를 통해 트레이딩에 유용한 신호로 변환된다. 예측된 가격 시계열로부터 수익률, 변동성, 선형 추세, 평균회귀, 모멘텀의 다섯 가지 신호를 체계적으로 추출하여, 각 예측 기간별로 5개씩 총 15차원의 종합적인 시장 신호 벡터를 형성한다. 이는 에이전트 상태와 함께 융합되어 최종적인 트레이딩 결정에 활용된다.

G. 학습 설계

학습 사이클 구성



[그림 13] 학습 사이클 종류

학습을 위해 4가지 사이클을 구성했다.

A. Non Episodic Stream Learning : 에피소드 길이 제한을 두지 않는다. PPO 온라인 업데이트를 위해 일정 스텝마다 멈춰서 신경망을 업데이트한다. 에

피소드가 파산, 만기일과 같은 상황으로 인해 종료되면 처음으로 돌아가지 않고 이어서 학습한다.

B. Non Episodic Survival Learning : (A)와 마찬가지로 에피소드 길이 제한을 두지 않는다. 파산 시에는 전체 학습 데이터의 시작점으로 돌아가 다시 학습한다.

C. Episodic Period-Based Learning : 에피소드 길이 제한을 둔다. 에피소드가 파산, 만기일과 같은 상황으로 인해 종료되면 처음으로 돌아가지 않고 이어서 학습한다.

D. Episodic Random Sampling Learning: 에피소드 길이 제한을 둔다. (C)와 달리 사전에 에피소드 길이만큼 데이터를 슬라이딩 윈도우 방식으로 잘라 보관해둔다. 환경을 초기화할 때마다 새로운 에피소드 데이터를 가져와 에피소드와 에피소드 사이의 시간 종속성을 없앤다.

(A), (C)는 실제 시장 상황과 가장 유사하다. 하지만 학습에 실패했을 때 다시 과거 시점으로 돌아갈 수 없기 때문에 데이터 효율이 떨어질 위험이 있다. (B)는 파산 방지를 최우선으로 하는 학습 사이클이다. 다만 파산에 큰 페널티를 부여하다 보니 학습 효율이 떨어지고, 목표인 수익 극대화를 배우기 어려울 수 있다. (D)는 에피소드 사이의 시간 종속성을 없앨 수 있지만 과적합이나 소극적인 행동 전략을 학습할 위험이 있다.

Non Episodic 방법론인 (A)와 (B) 은 에피소드 길이가 길어짐에 따라 분산이 커지는 자산 State의 특성상 아예 파산을 하지 않고 자산 운용을 배우는 것이 어려웠다. 특히 (B)의 경우, 파산을 피하다가 아예 거래를 포기해 손실을 내지 않는 모습이 빈번하게 포착되었다. 이 문제를 해결하기 위해 Episodic 방법론을 구상했다. (C)에서 부족한 데이터의 효율성을 높인 (D)는 예상과 달리 과적합 문제에서 자유롭지 못했으며, 소극적인 투자 전략으로 수렴했다. 따라서 최종적으로 (C)를 학습에 이용했다.

손실 함수 구성

$$L_{total} = L_{clip} + L_{value} + L_{entropy} + L_{entry\_reg}$$

1. L_{clip} : Clipped Surrogate Loss (정책 업데이트)
2. L_{value} : Value Loss (가치 함수 업데이트)
3. $L_{entropy}$: Entropy Bonus (탐험 증진)
4. L_{entry_reg} : Entry Regularization (진입 방향 정규화)

1. 정책 업데이트

기댓값을 최대화를 하기 위해 L_{clip}를 역수를 취한 후 줄인다.

$$L_{clip} = -E_t[min(r_t(\theta)\,A_t\,, clip(\,r_t(\theta), 1-\epsilon, 1+\epsilon\,)\,A_t)]$$

- $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_\theta old(a_t|s_t)}$: 현재 정책과 이전 정책의 확률 비율
- A_t : GAE(Generalized Advantage Estimation)로 계산된 advantage 값
- ϵ: 클리핑 하이퍼파라미터

2. 가치 함수 업데이트

Critic의 예측값과 실제 보상 간의 차이를 최소화한다.

$$L_{value} = \lambda_{value} \bullet MSE(V(s_t), R_t) = \lambda_{value} \bullet \frac{1}{N}\sum N(V(s_i) - R_i)^2$$

- $V(\,s_t\,)$: 상태 s_t에서 모델이 예측한 가치
- R_t : 실제 보상 (여기서는 GAE로 계산된 discounted return)
- λ_{value} : 가치 손실에 대한 가중치

3. 탐험 증진

정책의 엔트로피를 최대화해서 탐험을 장려한다. 엔트로피가 높을수록 정책의 랜덤성이 증가한다.

$$L_{entropy} = -\lambda_{entropy} \bullet E_t[H(\pi_\theta(\cdot|\,s_t))] = -\lambda_{entropy} \bullet E_t[-\sum\pi_\theta(a\,|\,s_t)log\pi_\theta(a\,|\,s_t)]$$

- $H(\pi_\theta(\cdot|\,s_t))$: 현재 정책 π_θ의 엔트로피
- $-\lambda_{entropy}$: 엔트로피 항에 대한 가중치

4. 진입 방향 정규화

진입 방향(매수/매도)에 대한 정책 분포를 특정 타겟 분포에 가깝게 만든다. KL Divergence를 이용해 두 분포 사이의 거리를 측정하고, 이를 최소화한다.

$$L_{entry\_reg} = \lambda_{entry} \cdot E_t[w_t \cdot KL(\pi_{entry\_current} \mid \pi_{entry\_target})]$$

- $\pi_{entry\_current}$: 현재 정책 분포

$$\pi\_(entry\_current) = \left(\frac{\sum_{j=1}^{K/2-1}\pi(a_j|s_t)}{\sum_{j=1}^{K}\pi(a_j|s_t)}, \frac{\sum_{j=K/2+1}^{K}\pi(a_j|s_t)}{\sum_{j=1}^{K}\pi(a_j|s_t)}\right)$$

- K : 전체 행동의 수 (e.g., 롱/숏/홀드)
- $\sum_{j=1}^{K/2-1}\pi(a_j|s_t)$: 숏 포지션에 해당하는 행동들의 확률 합
- $\sum_{j=K/2+1}^{K}\pi(a_j|s_t)$: 롱 포지션에 해당하는 행동들의 확률 합
- $\pi_{entry\_target} = (1 - p_{mix}, p_{mix})$: 타깃 분포
- $p_{mix} = \beta \cdot 0.5 + (1-\beta) \cdot \text{sigmoid}(\kappa \cdot score_t)$
- $score_t$: 상태별 트렌드 점수
- β: 유니폼 분포(0.5)와의 혼합 비율
- κ: 트렌드 점수 민감도
- $score_t = \frac{r_t + \mu_t}{\sigma_t + \epsilon}$

$$s.t.\ r_t = logReturn_t, \mu_t = rolling\ mean_t, \sigma_t = rolling\ std$$

트렌드 점수는 Z-score와 유사한 기능을 수행한다. 이 식은 현재 r_t이 과거 평균 수익률(롤링 평균)으로부터 표준편차의 몇 배만큼 떨어져 있는지를 나타내 이 값이 얼마나 이례적인지를 보여준다. 따라서 이 값이 양수이고 절대값이 크다면, 현재 수익률이 과거 평균보다 매우 높다는 뜻으로 해석할 수 있으며 반대로 음수이고 절대값이 크다면, 현재 수익률이 과거 평균보다 매우 낮음을 의미한다.

- $w_t = sigmoid(-regulation \cdot |\ score_t\ |)$:

가중치, 트렌드 점수 $score_t$가 클수록 규제가 완화된다.

- $KL(P \mid Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$

-λ_{entry} : 진입 정규화 항에 대한 가중치

진입 방향 정규화는 엔트로피만으로 충족되지 않는 진입 방향에 대한 탐험을 높인다. 청산 후 포지션 진입마다 탐험을 높이되, 트렌드에 반하지 않도록 규제를 추가했다.

학습 - 테스트 데이터 구성

[표 7] 학습-테스트 데이터 타임라인

학습 구간	테스트 구간
2010.02.17 - 2010.09.17	2010.09.20 - 2010.10.15
2010.10.18 - 2011.05.23	2011.05.24 - 2011.06.16
2011.06.18 - 2012.01.18	2012.01.19 - 2012.02.14
2012.02.15 - 2012.09.18	2012.09.19 - 2012.10.15
2012.10.16 - 2013.05.22	2013.05.24 - 2013.06.17
2013.06.18 - 2014.01.23	2014.01.24 - 2014.02.19
2014.02.20 - 2014.09.29	2014.09.30 - 2014.10.24
2014.10.27 - 2015.06.03	2015.06.04 - 2015.06.26
2015.06.29 - 2016.02.01	2016.02.02 - 2016.02.29
2016.03.02 - 2016.10.07	2016.10.10 - 2016.11.01
2016.11.02 - 2017.06.09	2017.06.12 - 2017.07.04
2017.07.05 - 2018.02.12	2018.02.13 - 2018.03.12
2018.03.13 - 2018.10.22	2018.10.23 - 2018.11.14
2018.11.15 - 2019.06.25	2019.06.26 - 2019.07.18
2019.07.19 - 2020.03.10	2020.03.11 - 2020.04.03

다양한 시기에 robust하게 대응하는 트레이딩 에이전트를 개발하기 위하여, 학습 구간과 테스트 구간을 다음과 같이 설정했다. 총 15개의 구간으로 전체 데이터를 나눈 후, 9:1 비율로 학습과 테스트 구간을 나눴다.

VI. 결론 및 시사점

A. 결과 분석

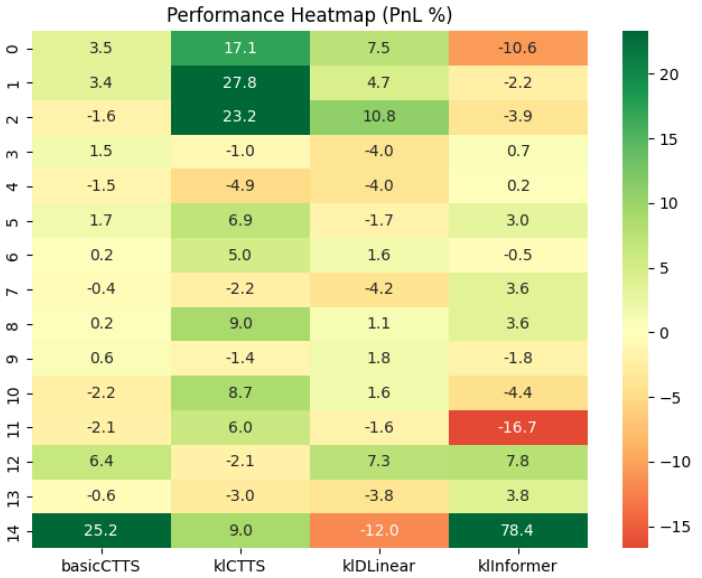
확률적으로 전이되는 에이전트의 정책의 특성상, 테스트 상황에서는 결정론적 선택이 일반적이다. 하지만 우리 모델은 hold의 확률이 다소 높아 결정론적인 선택을 적용하기 어려웠다. 따라서 확률 정책의 robust함을 보이기 위해 동 구간에서 에피소드를 30번 씩 반복한 뒤 평균 결과를 구했다.

[표 8] 테스트 구간에서 P&L (%)

테스트 기간	CTTS(Basic)	CTTS(KL)	DLinear(KL)	Informer(KL)
2010.09.20 – 2010.10.15	3.5	**17.1**	7.5	−10.6
2011.05.24 – 2011.06.16	3.4	**27.8**	4.7	−2.2
2012.01.19 – 2012.02.14	−1.6	**23.2**	10.8	−3.9
2012.09.19 – 2012.10.15	**1.5**	−1.0	−4.0	0.7
2013.05.24 – 2013.06.17	−1.5	−4.9	−4.0	**0.2**
2014.01.24 – 2014.02.19	1.7	**6.9**	−1.7	3.0
2014.09.30 – 2014.10.24	0.2	**5.0**	1.6	−0.5
2015.06.04 – 2015.06.26	−0.4	−2.2	−4.2	**3.6**
2016.02.02 – 2016.02.29	0.2	**9.0**	1.1	3.6
2016.10.10 – 2016.11.01	0.6	−1.4	**1.8**	−1.8
2017.06.12 – 2017.07.04	−2.2	**8.7**	1.6	−4.4
2018.02.13 – 2018.03.12	−2.1	**6.0**	−1.6	−16.7
2018.10.23 – 2018.11.14	6.4	−2.1	7.3	**7.8**
2019.06.26 – 2019.07.18	−0.6	−3.0	−3.8	**3.8**
2020.03.11 – 2020.04.03	25.2	9.0	−12.0	**78.4**
평균 (%)	2.29	**6.5**	0.3	4.07
총합 (원)	10,300,155	**29,400,454**	1,518,325	18,347,792

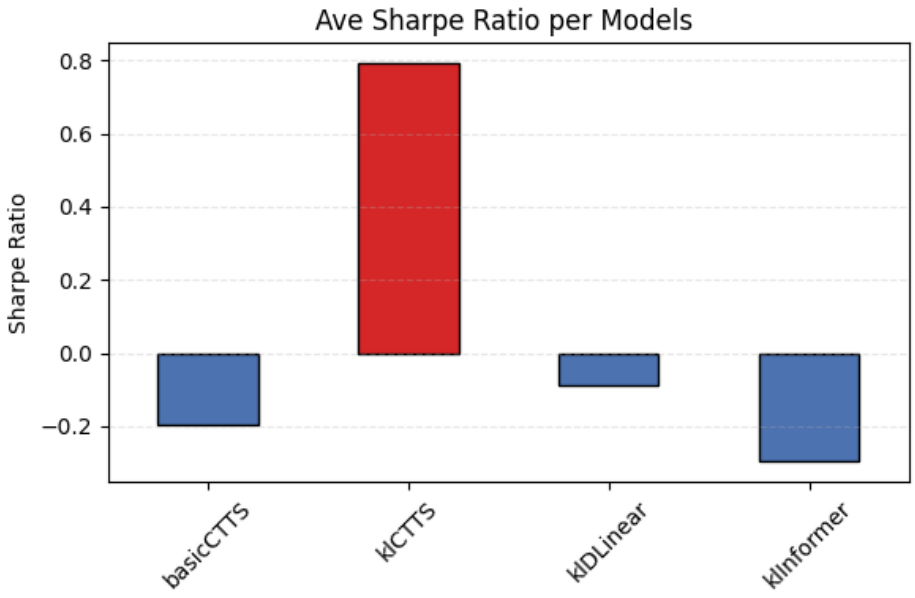
순서대로 CTTS에다가 KL 진입 방향 정규화를 추가하지 않은 모델, CTTS KL

진입 방향 규제 포함 모델, DLinear KL 추가 모델, Informer KL 추가 모델이다.



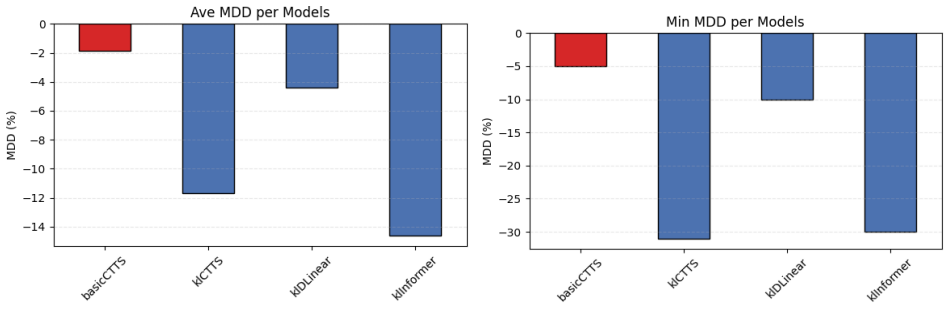
[그림 14] 수익률 히트맵

모든 모델에서 수익이 발생했다는 점에서 위에 제시된 방법론들의 우수성을 입증할 수 있다. 세부적으로는 KL 진입 방향 규제를 추가한 모델이 추가하지 않은 모델보다 성능이 뛰어났다. CTTS에 KL을 추가한 모델은 다른 모델들보다 안정적인 수익률을 보였으며, 손실 빈도와 규모가 다른 모델들에 비해 낮았다.



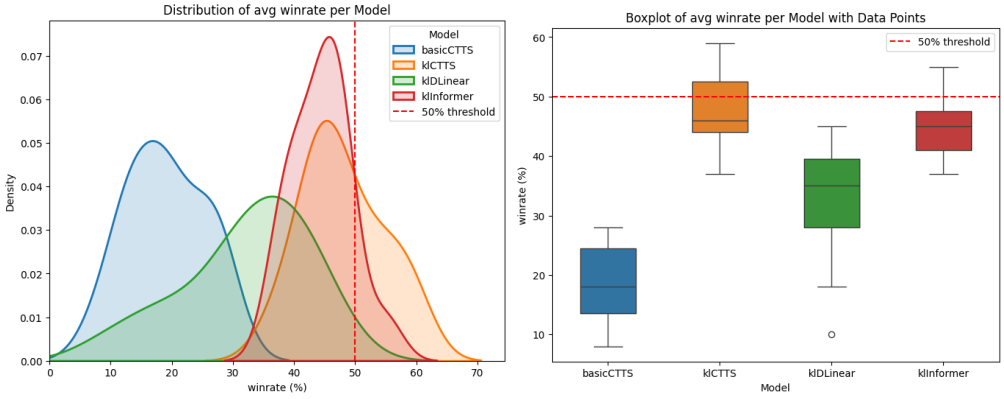
[그림 15] 모든 에피소드의 평균 Sharpe Ratio

Sharpe Ratio는 투자의 위험 대비 수익을 측정하는 지표로 투자 성과를 평가할 때 사용된다. 일반적으로 0을 넘어야 투자의 대상이 될 수 있으며, 0.5-1면 양호한 전략, 1을 넘는 전략은 우수한 전략이라 평가한다. 본 논문에서 구현한 4개의 모델 중 Sharpe Ratio가 1을 넘는 모델은 CTTS에 KL 항을 포함한 모델이며, Sharpe Ratio가 0.8로 양호한 수치를 띈다.



[그림 16] 모든 에피소드의 평균 MDD / 최저 MDD

MDD(Max Drawdown)은 포트폴리오 가치의 최고점에서 최저점 사이 하락폭을 측정한다. 이는 투자 시장에서 잠재적인 손실을 모델링할 때 유용하다. MDD의 경우, 수익과 손실의 절대량이 적은 KL 항이 없는 CTTS가 가장 우수했고, 그 다음 비슷한 양상을 띄는 KL 항을 추가한 DLinear가 다음으로 우수했다. KL 항이 있는 CTTS와 Informer는 최대 최저 수익률을 미루어 비춰보아 강한 거래 체결을 하기 때문에 더 작은 MDD를 보인다.



[그림 17] 모든 에피소드의 평균 승률 (%)

승률은 청산 시점에서 이득을 보았는가를 측정하는 지표다. KL항을 추가하지 않은 CTTS는 평균 승률 19%이며, 최고 승률은 40%를 넘지 못했다. KL항을 추가하지 않은 CTTS보다는 전체적으로 좀 더 높은 승률 분포를 갖고 있지만, DLinear를 기반으로 하는 모델도 평균 승률이 32%로 미미했다. 4개의 모델

중 가장 승률이 50%에 가까운 KL 항을 포함하는 CTTS는 평균 48%로 높은 승률을 보였으며, 최대 70%까지 높은 승률을 보인다. 그 다음으로 따라가는 Informer는 평균 44%의 승률을 보였으며, CTTS보다 더 분산이 적은 모습을 보였다.

가장 높은 안정성을 띄며 수익을 내는 모델은 KL로 진입 방향 규제를 시행한 CTTS이며, **평균 수익률 : 6.5% , Sharpe Ratio : 0.8 , MDD : -12% , Win Rate : 48%**를 기록했다. 본 모델은 테스트 구간에서 긍정적인 수익성과 효율적인 위험 관리 능력을 동시에 보여주었다.

B. 시사점

본 논문에 제시된 모델은 수익성과 안정성을 균형 있게 확보한 투자 전략으로서의 잠재력을 가지고 있으며, 실제 투자 환경에 적용될 수 있는 중요한 가치를 지닌다. 특히 실제 미니 KOSPI 200 선물 수수료와 슬리피지를 고려한 모델임에도, 높은 Sharpe Ratio와 낮은 MDD는 모델이 단순히 수익률을 높이는 것을 넘어, 시장 변동성 위험을 효과적으로 관리하고 있음을 보여준다. 이러한 강점은 불확실성이 큰 실제 선물 시장 환경에서 더욱 빛을 발할 것이다.

C. 추가 연구 및 발전 방향

향후 연구에서 모델의 성능을 더욱 향상시키고, 실제 적용 가능성을 높이기 위해 다음과 같은 방향을 구상했다.

1. 결정론적 정책을 위한 hold 비율 최적화

강화학습 모델은 확률적 행동(Stochastic Policy)을 기반으로 하지만, 실제 투자 및 테스트에서는 예측 가능한 행동(Deterministic Policy)이 더 안전하다. 현재의 정책은 hold 확률이 높아 결정론적 정책에서 유의미한 결과를 내지 못한다. 이 문제를 해결하기 위해 hold 포지션의 확률을 효과적으로 제어하는 학습 방습을 도입하여, 시장 결정론적 행동이 가능케 만들어 보고자 한다.

2. 다수결 앙상블을 통한 견고성 및 성능 향상

단일 모델이 아닌 여러 모델의 결과를 종합하는 앙상블 기법을 적용하고자 한다. 예를 들어 다른 랜덤 seed나 하이퍼 파라미터로 학습된 여러 모델의 예측을 결합한다거나, 구현한 다양한 모델의 예측을 결합하여 최종 행동을 다수결 투표 방식으로 결정하는 것이다. 이러한 다수결 앙상블은 개별 모델의 편향이나 오류를 상쇄하여 예측의 견고성을 크게 강화하고, 전반적인 성능을 더욱 안정적으로 끌어올릴 수 있다.

참고문헌

He, J., Zheng, C., & Yang, C. (2023). Integrating tick-level data and periodical signal for high-frequency market making. arXiv preprint arXiv:2306.17179.

Liu, X.-Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., & Wang, C. D. (2020). FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. arXiv preprint arXiv:2011.09607.

Mou, S., Xue, Q., Chen, J., Takiguchi, T., & Ariki, Y. (2025). MM-iTransformer: A multimodal approach to economic time series forecasting with textual data. Applied Sciences, 15(3), 1241. https://doi.org/10.3390/app15031241

Sadighian, J. (2019). Deep reinforcement learning in cryptocurrency market making. arXiv preprint arXiv:1911.08647.

Sadighian, J. (2020). Extending deep reinforcement learning frameworks in cryptocurrency market making. arXiv preprint arXiv:2004.06985.

Sood, S., Papasotiriou, K., Vaiciulis, M., Balch, T. H., & Morgan, J. P. (2023). Deep reinforcement learning for optimal portfolio allocation: A comparative study with mean-variance optimization. (unpublished manuscript/preprint).

Tang, C. Y., Liu, C. H., Chen, W. K., & You, S. D. (2020). Implementing action mask in proximal policy optimization (PPO) algorithm. ICT Express, 6(3), 200–203. https://doi.org/10.1016/j.icte.2020.05.003

Wang, L., Chen, Y., Yu, G., Li, S., & Wu, X. (2024). A closer look at invalid action masking in policy gradient algorithms. Electronics, 14(16), 3327. https://doi.org/10.3390/electronics14163327

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In Proceedings of the AAAI Conference on Artificial Intelligence, 37(10), 11118–11126.

Zeng, Z., Kaur, R., Siddagangappa, S., & Rahimi, S. (2023). Financial time series forecasting using CNN and Transformer. arXiv preprint arXiv:2304.04912.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. arXiv preprint arXiv:2012.07436.

Zhou, H. (2020). Informer2020 (GitHub repository). Retrieved from https://github.com/zhouhaoyi/Informer2020

KOSPI200 주가지수 선물/옵션 매매제도. (n.d.). Kiwoom Securities. Retrieved from https://www.kiwoom.com/h/help/trade/VHelpFuopTradeSystemView

선물 (금융). (2025년 4월 7일). 위키백과. Retrieved from https://ko.wikipedia.org/wiki/%EC%84%A0%EB%AC%BC_(%EA%B8%88%EC%9C%B5)

미니 코스피200선물. (n.d.). Eugene Investment & Futures Co., Ltd. Retrieved from https://www.eugenefutures.com/main/IG/view/IG_0102_T5P2.htm?prdtId=IG_0102_T5P2

시사경제용어사전. (n.d.). Ministry of Economy and Finance. Retrieved from https://www.moef.go.kr/sisa/main/main

선물옵션 거래안내. (n.d.). Samsung Futures. Retrieved from https://www.samsungpop.com/?MENU_CODE=M1568700777221