강화시스터즈 3기 03세션

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

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

목차

I. 서론

최근 글로벌 금융시장은 코로나19 팬데믹, 지정학적 불안정성, 중앙은행 정책 변화 등으로 인해 높은 변동성을 보이고 있다.
특히 한국의 **KOSPI200 선물시장**은 1일 평균 거래대금이 10조원을 넘는 대형 시장임에도, 기관투자자와 개인투자자 간의 **정보 비대칭**과 **고빈도 거래 확산**으로 인해 전통적 분석 방법의 한계가 두드러진다.

강화학습의 장점

- 불확실·동적 환경에서 **연속적인 의사결정**을 통해 수익 추구 가능
- 지도학습: 과거 패턴 학습 → 미래 예측
- 강화학습: 시장 상황에 따라 **매수·매도·관망 행동 직접 학습**
  → 실제 트레이딩 환경에 적합한 접근법 제공

연구의 차별성

- **실거래 수준의 데이터 적용** → KOSPI200 미니 선물 분봉 데이터
- **행동 중심 학습** → 거래 강도·자산 운용 전략까지 학습
- **보상 설계 확장** → 위험조정 성과까지 고려한 보상 함수 도입
- **다양한 신경망 구조 비교** → DLinear, CNN+Transformer, Informer

연구 목표 1

PPO 알고리즘 기반 강화학습 에이전트가
KOSPI200 선물 시장에서
유의미한 수익률을 달성할 수 있는지 검증

연구 목표 2

DLinear, CNN+Transformer, Informer 등
서로 다른 신경망 구조 중 어떤 것이 금융 시
계열 데이터의 특성을 효과적으로 학습하는
지 비교

연구 목표 3

누적 수익률, 위험 조정 수익률 등을
극대화하기 위한 다양한 형태의
보상함수를 설계하고 그 영향력을 분석

II. 배경 지식 - 선물 시장

선물은 파생상품의 한 종류로 품질, 수량, 규격 등이 표준화되어있는 상품 또는 금융자산을 미리 결정된 가격으로 미래 일정시점에 인도・인수할 것을 약정한 거래. 본 프로젝트에서는 그 중 **미니 코스피200 주가지수 선물**을 대상으로 함.

미결제약정 (Open interest)

투자자가 보유하고 있는 계약

증거금 (Margins)

선물 계약 이행을 위한 보증금
- 위탁증거금: 거래 시 최초 납부해야하는 증거금 (10%)
- 유지증거금: 포지션 유지를 위해 계좌에 최소 유지해야하는 증거금 (7%)

마진콜 (Margin call)

거래 중 발생한 손실로 계좌의 증거금 수준이 유지증거금 이하로 떨어졌을 때, 증권사가 투자자에게 추가증거금을 채워넣으라고 요구하는 것

만기일 (Maturity)

기초자산의 인도가 약속된 날.
금융 자산의 경우 직접 인도가 아닌 **청산**으로 실현됨.

포지션 (Position)

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

일일정산 (Daily settlement)

매일 장 종료 시의 가격에 따라 현재 투자자가 보유 중인 미결제약정에 대해 증거금 계좌의 변동이 이루어지는 것

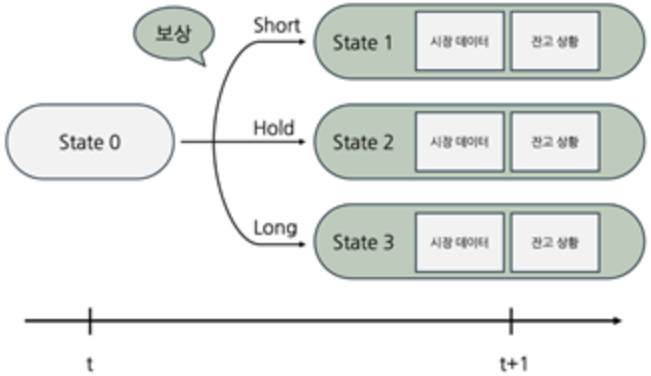
II. 배경 지식 - 선물 시장의 MDP

MDP 정의

- 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
- $Action\ Space$: $\{-k, \cdots, -1, 0, 1, \cdots, k\}\ s.t.\ k=10$

행동의 절대값은 계약 수, 부호는 각기 숏(-)과 롱(+)을 의미. 0은 홀드

선물 시장에서 행동에 따른 상태 전이



금융 시계열 데이터의 MDP는 어떤 행동을 선택하더라도 State가 전이된다는 특징이 있다.

에이전트가 Hold를 선택하더라도 시장가는 변하기 때문이다.

II. 배경 지식 - 시계열 데이터

시계열 데이터 : 하나의 변수를 시간에 따라 여러 번 관측한 데이터로, 일정한 시간 간격으로 수집된 연속적인 관측값들을 의미

➔ 금융 분야에서는 가격, 거래량, 변동성 등의 변화 예측과 반복되는 패턴에 대한 인사이트 도출을 목표로 함

금융 시계열 데이터의 특성

- 높은 변동성 - 변동성 군집화 현상
- 두꺼운 꼬리 분포 - 극단적 값들이 예상보다 자주 발생
- 레버리지 효과 - 가격 하락 시 변동성이 상승 시보다 더 크게 증가
- 비정상성 - 평균과 분산이 시간에 따라 변화

시계열 데이터 구성 요소

- 추세 - 장기적인 증가 또는 감소 패턴
- 계절성 - 특정 요일이나 계절에 따라 일정한 주기로 반복되는 패턴
- 주기성 - 고정된 빈도가 아니지만 형태적으로 유사하게 나타나는 패턴
- 노이즈 - 측정 오류나 내부 변동성 등 다양한 요인으로 생기는 불규칙적인 변동

전처리 과정

결측치 처리

전진 대체법(Forward Fill, ffill)이 데이터의 연속성을 유지하는 데 효과적

정규화

Min-Max 정규화나 Z-score 표준화를 통해 서로 다른 스케일의 변수들을 동일한 범위로 조정

정상성 확보

차분이나 로그 변환 등을 사용하여 비정상 시계열을 정상화

III. 핵심이론 - PPO

PPO(Proximal Policy Optimization) 알고리즘

| TRPO(Trust Region Policy Optimization, 신뢰 영역 정책 최적화) 알고리즘의 복잡한 제약 최적화 문제를 피하고,
정책 업데이트의 안정성을 높인 알고리즘 . >> 이전 정책과 현재 정책 사이의 거리를 유지시키기

대리 목적 함수와 확률 비율

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

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

정책 기반 알고리즘의 최적화 대상 J 대신 대리목적 함수 L^CPI를 최적화한다.
확률 비율은 신규 정책과 이전 정책 사이의 변화를 측정한다.

클리핑 오류 함수

$$L^{CLIP}(\theta)$$

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

PPO의 핵심으로, 기본 목적 함수에서 나타나는 A가 양수일 때 r이 무한정으로 늘어나는 경향을 억제한다. 정책 성능의 하한선의 역할을 수행하며, 과도한 변화를 무시한다.

GAE

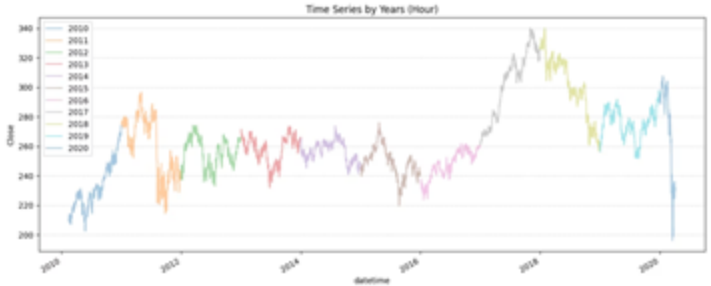
$$\hat{A}_t^{GAE(\gamma,\lambda)} = \sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+l}$$

에피소드의 길이가 다 다를 때 일반적인 어드벤티지 추정치를 가중 평균을 통해 제어한다. 두 개의 하이퍼파라미터 할인율 γ와 평활화 파라미터 λ에 의해 제어된다.

Ⅳ. 데이터 분석 및 전처리

데이터 개요

사용된 데이터는 2010년 2월부터 2020년 4월까지 약 10년간의 **코스피 200 미니 선물 거래 정보**다.
1분 단위의 시가, 고가, 저가, 종가, 거래량 데이터를 포함하고 있으며, 특히 2020년 데이터에는 COVID-19 팬데믹 기간의 극심한 시장 변동성이 반영되어 있다.



1 2010-2011 초기 회복기
금융위기의 여파에서 벗어나며 점진적인 상승세 관찰

2 2012-2015 안정기
큰 변동 없이 완만한 등락을 반복하는 횡보장이 지속

3 2016-2018 상승기
국내외 경제 회복과 유동성 공급 확대의 영향으로 뚜렷한 상승 모멘텀이 나타났으며, 전체 분석 구간에서 가장 강력한 불장을 형성

4 2019-2020 변동기
미중 무역갈등과 글로벌 불확실성이 커지면서 시장 분위기가 바뀌기 시작했고,
2020년 초 코로나19 팬데믹 충격으로 인해 급격한 폭락과 극심한 변동성을 경험한 후 빠른 반등이 이어짐

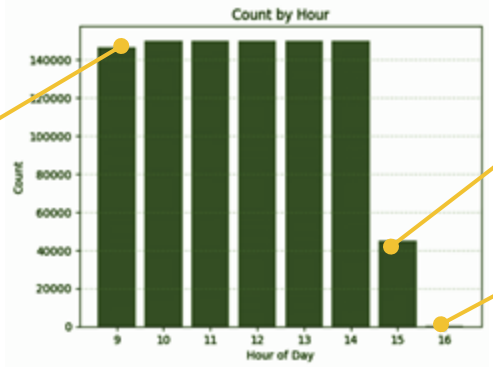
이처럼 안정적인 구간부터 극한 상황까지 포함하는 다양한 시장 환경은 강화학습 모델이
일반적인 거래 상황뿐만 아니라 위기 대응 능력까지 함께 학습할 수 있는 이상적인 데이터셋을 제공한다.

데이터 특성

시간대별 데이터 분포 특성

1 **9시 데이터**

매년 수능일·신정(1/2) → 개장 1시간 지연



2 **15시 데이터**

상대적으로 데이터 수가 부족

장 마감 시간 변경 영향 (2016.8 이전: 15:45 → 이후: 15:30)

3 **16시 데이터**

수능으로 인한 장 개폐 지연

2017년 포항 지진 → 수능 연기, 장 마감 16시 (2회)

데이터 전처리

기본 전처리

시스템 지연 데이터

- 15:06 → 15:05, 16:06 → 16:05로 통합

결측치 처리

- 전체 타임스탬프 연결 → 누락 시점 NaN 패딩
- 장 마감 직전 10분 구간 → 데이터 공백 보완 (fill 적용)

특이 데이터 처리

- 2010.07.16 → 15:15 데이터만 존재 (OHLC 불일치) → 분석 제외
- 2020.03.13 & 03.19 → 서킷브레이커 발생, 29개 결측치 → 보간 대신 제외

시간 기준으로 타임스텝 당 1-2개 수준의 소규모 결측치에 대해서는 "**마지막 가격이 유지된다**"는 시장 원리에 따라 전진 채움법(Forward Fill)을 적용

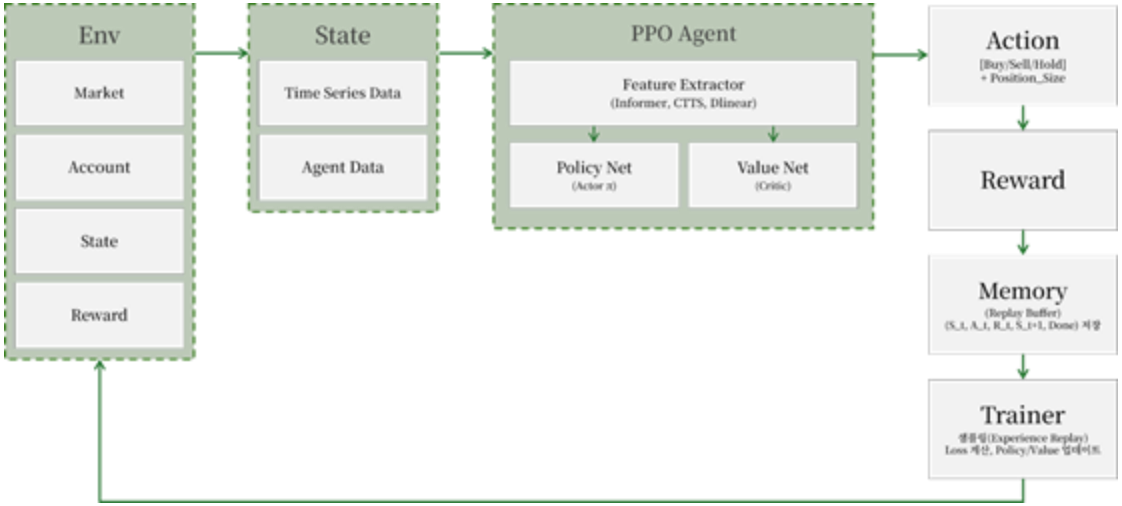
이는 급격한 가격 변동보다는 보수적 접근을 통해 모델의 안정성을 확보하기 위한 전략

핵심 컴포넌트 구현

환경, 에이전트, State와 같이 학습에 필수적인 컴포넌트를 설계

프로젝트 구조도

본 프로젝트는 환경, 신경망, 강화학습 에이전트, 트레이닝 등 여러 모듈로 구성되어 있다.
각 모듈은 유기적으로 연결되어 선물 시장에서의 트레이딩 전략을 최적화한다.



환경 모듈

시장·계좌 데이터 관리, 상태·보상 제공

신경망 모듈

DLinear·CTTS·Informer 기반 특징 추출

에이전트 모듈

PPO 기반 강화학습, 매수/매도/홀드 결정

트레이닝 모듈

경험 메모리, 리플레이 버퍼, 정책 업데이트

이 구조도는 데이터 흐름과 각 컴포넌트 간의 상호작용 시각적으로 나타냄.
신경망 모델은 시장 데이터를 분석하여 유용한 특징을 추출하고, 강화학습 에이전트는 이를 바탕으로 최적의 트레이딩 결정을 내림.

환경 구현: 선물 시장 및 계좌 관리

선물 시장 제어 클래스

- 강화학습의 메인 환경 역할
- 현재 상태에서 행동을 받아 다음 상태, 보상, 종료 여부 계산
- 선물 계약 단위, 만기일, 거래 비용, 슬리피지 등 관리
- 에이전트의 행동 제약 및 성과 평가

계좌 클래스

- 투자자의 계좌, 포지션, 손익 관리
- 계약 체결 및 청산 기능
- 위탁 증거금 및 유지 증거금 관리
- 실현/미실현 손익 계산
- 일일정산 구현

환경 동작 과정

01.
매 step마다 다음 데이터와 에이전트의 행동에 따라 시장 상태와 계좌 업데이트

02.
종료 조건 및 강제 청산 여부 계산, 거래 결과 저장

03.
에이전트의 성과 추적 및 보상 계산

04.
다음 state 생성 및 행동 제약 계산

본 연구에서 에이전트가 학습할 매매 전략의 근간은 **스윙 트레이딩(Swing Trading)**
단기적인 가격 변동이 형성하는 추세의 전환점을 포착하여 수익을 창출하는 중단기 투자 방법론.

State 설계: 시장 정보와 에이전트 상태

강화학습 에이전트에게 제공되는 상태(State)는 시계열 데이터셋과 에이전트 정보가 결합된 형태로 설계됨

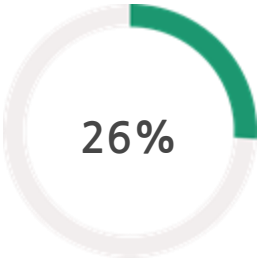
시계열 선물 시장 변수

- 가격 데이터: close, high, low
- 거래량: volume_change
- 추세 지표: ema_5, ema_20, ema_cross
- 모멘텀 지표: rsi, %K, %D, cci
- 변동성 지표: atr, bb_width
- 거래량 지표: obv

에이전트 변수

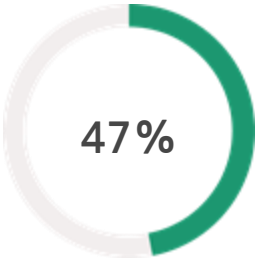
- current_position: 현재 포지션 상태
- excution_strength: 계약 체결 강도
- n_days_before_ma: 만기일까지 남은 일수
- realized_pnl: 실현 손익
- unrealized_pnl: 미실현 손익
- available_balance: 가용 잔고
- market_regime: 시장 국면 정보

시장 국면 분류



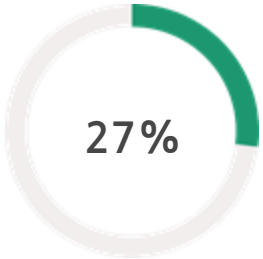
강세장

30틱 초단기 MA > 150틱 중단기 MA (1 + 0.001)



횡보장

30틱 초단기 MA < 150틱 중단기 MA * (1 - 0.001)



약세장

그 외 구간

보상 및 에피소드 설계

보상 함수 구성

최종 보상 함수 Rt는 수익(Rprofit), 위험(Rrisk), 후회(Rregret) 세 가지 요소의 가중합과 상황에 따른 패널티로 구성됨.

1

수익 구성 요소

실현 손익과 미실현 손익의 변화량을 로그 변환하여 계산

2

위험 구성 요소

Differential Sharpe Ratio(DSR)를 활용한 위험 조정 수익률 변화 함수. 포트폴리오 가치 변화량의 샤프 지수 미분값으로 계산.

3

후회 구성 요소

포지션을 보유하지 않았을 때 시장이 움직인 것에 대한 기회비용. 시장 가격 변화와 현재 포지션을 고려하여 계산

보너스 및 패널티

상황	보상 값
마진콜	-2.0
파산	-5.0
만기일	-0.5
목표 달성 (수익률 5% 초과)	+2.0

에피소드 설계

현실 시간에서 약 2주에 해당하는 3,000 스텝을 에피소드 최대 길이로 설정. 다양한 시장 상황에 따른 종료 및 청산 조건은 다음과 같음:

강제 종료 조건

마진콜, 파산, 만기일, 최대 길이 도달

청산 조건

마진콜, 파산, 만기일, 최대 길이 도달

특별 처리

당일 마지막 틱에서는 미실현 손익을 실현 손익으로 전환

보상 및 에피소드 설계

DSR

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

- 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

보상 및 에피소드 설계

수식

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

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

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

- $P_{N,t}$: 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

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

- D_t : 위험 조정 수익률 변화 함수 (Differential Sharpe Ratio, DSR)
- $P_{PV,t-1}$: 보유 자산 + 미실현 손익으로 구성된 포트폴리오 가치의 변화량

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

- ΔP_t : 시장의 가격 변화
- B_t : 2차 미분값, 분산
- η : EMA 업데이트를 위해 사용되는 파라미터

에이전트 구현: PPO 알고리즘

행동 공간 정의 및 제약 조건

본 연구의 에이전트는 선물 시장에서 능동적으로 진입 및 청산 시점을 관리할 수 있도록 설계함

행동 공간

- -10(10계약 매도)부터 +10(10계약 매수)까지의 정수 값
- 0은 포지션 유지(Hold) 의미

제약 조건

- 초기 자본금: 30,000,000 원
- 최대 포지션 상한: 10계약
- 현재 포지션에 따라 일부 행동이 제한될 수 있음

유효하지 않은 행동 처리 기법: 액션 마스킹



행동 무시
유효하지 않은 행동이 선택되면 이를 무시하고 이전 상태를 유지

페널티 부여
유효하지 않은 행동 선택 시 에피소드 종료 및 큰 음수 보상 부여

액션 마스킹
유효하지 않은 행동의 확률 값을 0으로 만들어 선택 가능성 배제

본 연구에서는 학습 효율성을 최우선으로 고려하여 **액션 마스킹** 방식을 채택 함. 이는 에이전트가 명백히 불가능한 행동을 탐험하는 데 시간을 낭비하지 않도록 하여 탐험 효율성을 극대화하고, 더 빠르고 안정적인 정책 학습을 가능하게 함

신경망 이론 & 액터 크리틱 구조

장기 시계열 예측에 효과적이라 알려진 세 가지 신경망 모델인

DLinear, CNN+Transformer Hybrid Model, Informer를 활용하여

선물 시장 트레이딩 시스템을 구현함

DLinear: 시계열 분해와 단순함의 힘

DLinear는 Transformer가 장기 시계열 예측(long-term TSF)에 정말로 효과적인지에 의문을 제기하며,
시계열 데이터의 특성을 살린 효과적인 모델을 제안함

Transformer의 한계

Transformer의 핵심인 multi-head self-attention은 시간 순서에 관계없이 작동하여 시계열 데이터의 시간 순서 정보를 무시할 수 있음.

DLinear의 단순한 구조

시계열 분해(decomposition)와 단순한 one-layer linear network를 결합한 구조로, 시계열 데이터의 특성을 효과적으로 활용함

DLinear의 장점

효율적인 신호 처리

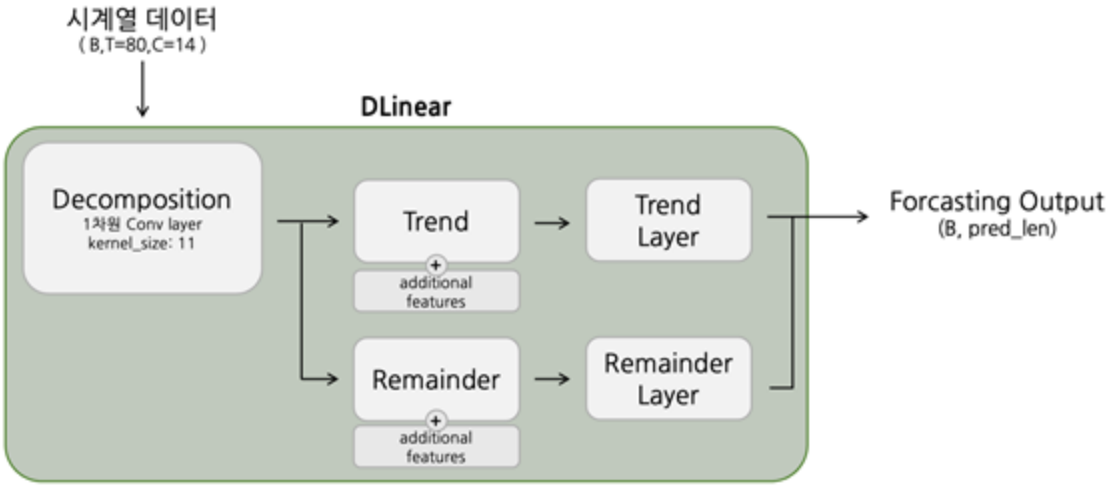
O(1) 최대 신호 경로 길이로 단기 및 장기 시간적 관계를 낮은 연산량으로 포착

해석 용이성

결과 해석이 용이하고 모델 하이퍼파라미터 튜닝이 거의 필요 없음

높은 효율성

one-Layer Linear network만 사용하여 메모리 효율이 좋고 처리 속도가 빠름



DLinear : 액터 크리틱 구조

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 (Decompose)

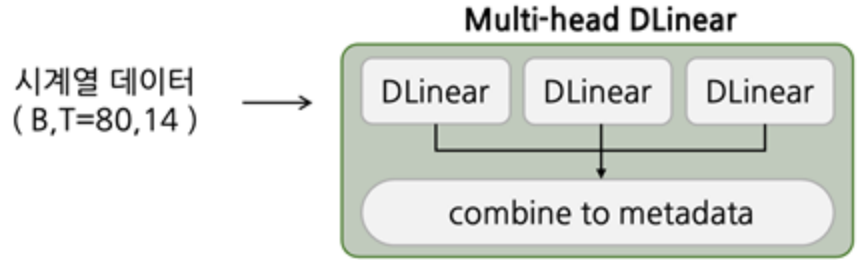
'close'가격에 대해서만 시계열 분해를 수행합니다.
슬라이딩 윈도우 방식으로 시계열 데이터의 Trend를 분리하고, 원본 데이터에서 Trend를 뺀 것이 Remainder.
1차원 합성곱 레이어 nn.Conv1d로 구현하였다.

단일 DLinear 모델

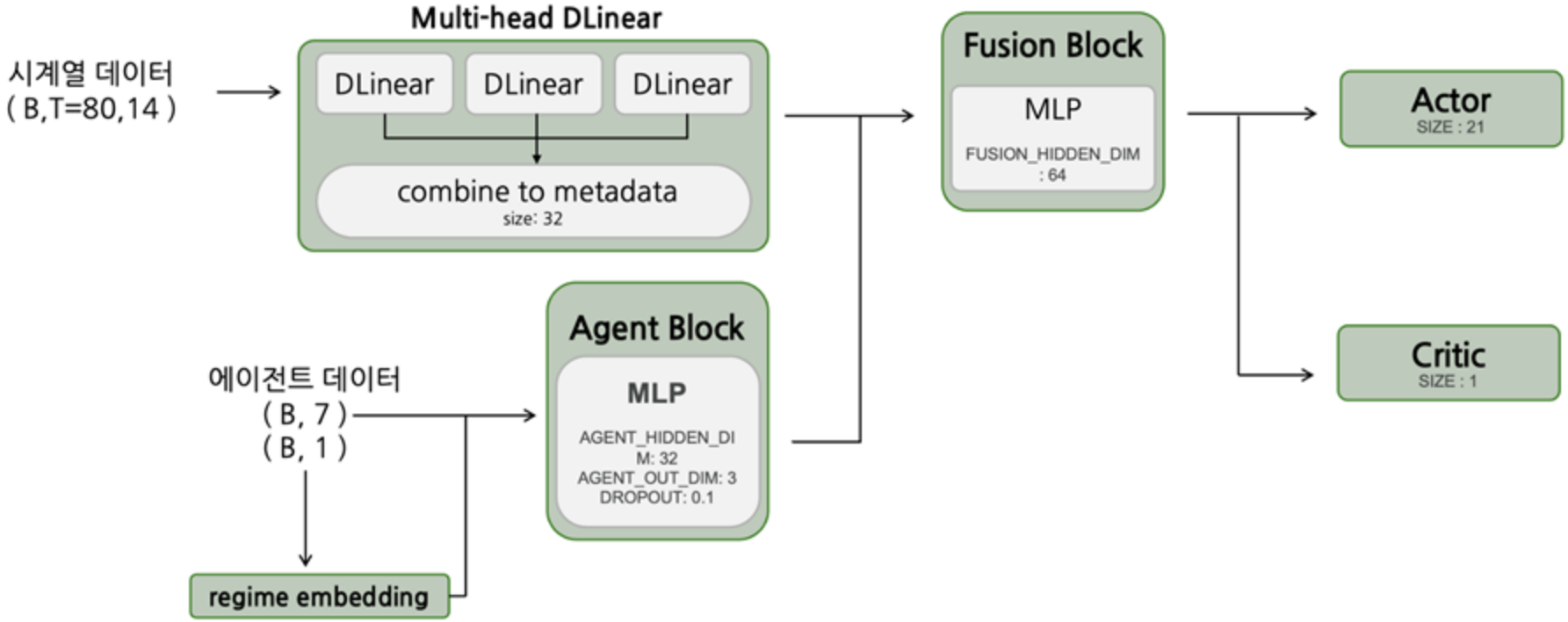
Trend와 Remainder 각각의 one-layer linear network를 갖는다. 분해한 결과에 다른 feature를 결합한 데이터를 각각의 linear layer에 통과시켜 최종 예측값을 출력한다.

Multi-head DLinear 모델

여러 개의 DLinear 모델이 서로 다른 예측 길이로 예측값을 출력하
모든 head의 결과를 종합해 최종 결과값을 얻는다.
단순 평균 방법과 메타데이터 계산 방법 두 가지로 구현했다.



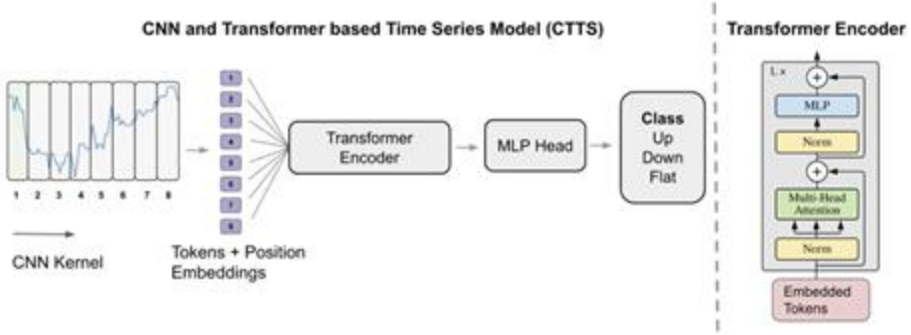
DLinear : 액터 크리틱 구조



CTTS:

CNN과 Transformer의 하이브리드 모델

시계열 데이터 분석의 핵심은 미세한 **지역적 패턴**과 **장기적 패턴**을 동시에 포착하는 것.



CNN의 강점

- 합성곱 커널을 이용한 지역적 패턴 추출
- 격자 형태의 행렬 데이터에서 공간적 정보 추출
- 시각적 데이터 처리에 효과적

Transformer의 강점

- Attention 메커니즘을 통한 장기 종속성 포착
- 입력 시퀀스 전역에 걸친 정보 처리
- 전역적 패턴 인식에 탁월

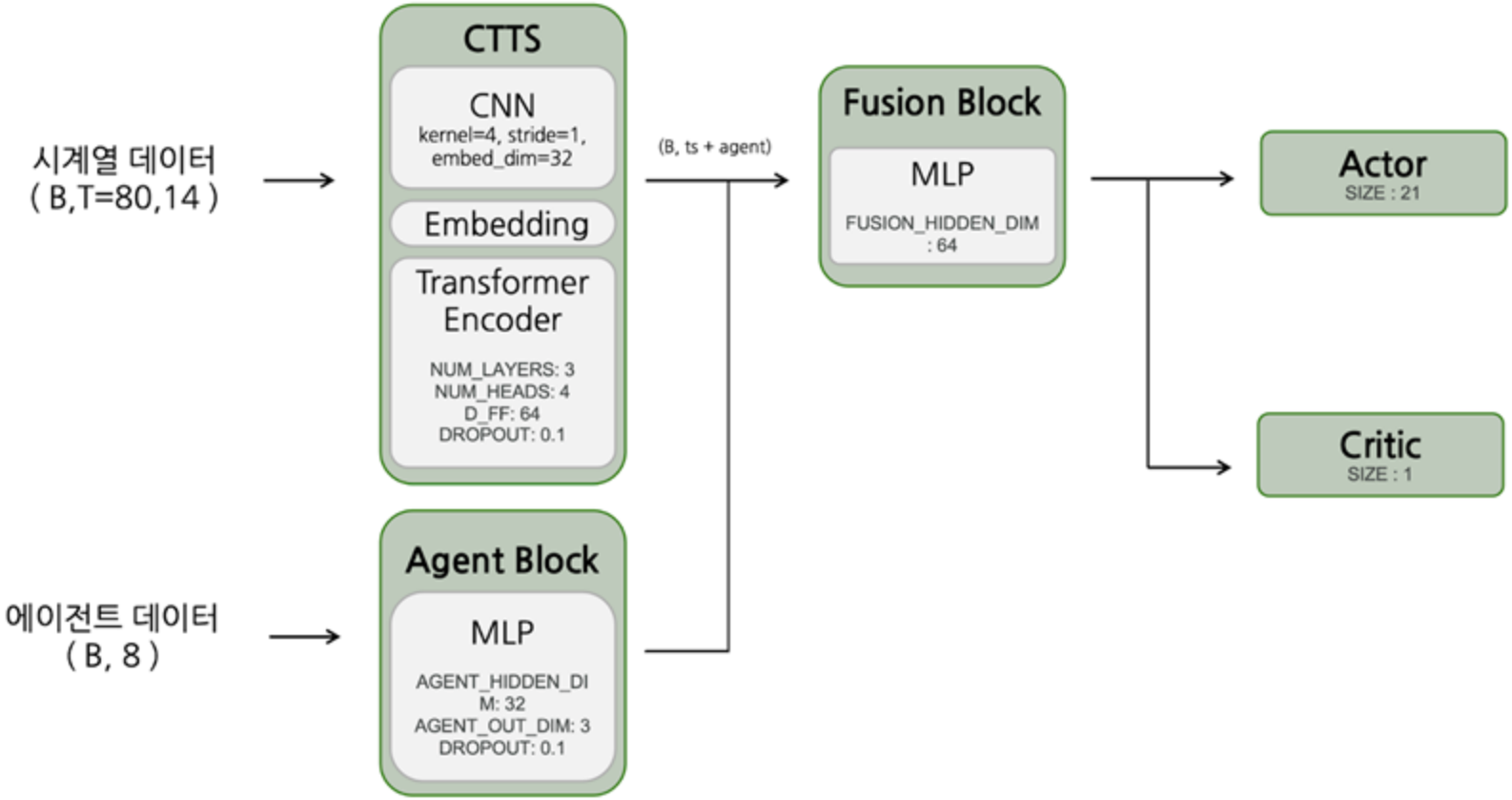
CTTS 모델은 Pre-LN 방식을 채택하여 그래디언트 안정성을 확보함.
초기 트랜스포머에게 필요했던 학습률 웜업 단계 없이도 원만한 학습을 가능하게 함

CTTS(CNN and Transformer based time series modeling)는

CNN이 지역적 패턴을, 트랜스포머 인코더가 장기적 패턴을 포착해 서로의 한계를 보완한다.

>> 시계열 데이터의 다양한 패턴을 학습할 수 있음 !

CTTS: 액터 크리틱 구조



Informer: 효율적인 장기 시계열 예측 모델

Informer는 기존 시계열 모델의 한계를 극복하고 효율성을 높이기 위해 개발

기존 모델의 한계

RNN: 기울기 소실 문제로 장기 의존성 학습에 한계

CNN: 고정된 필터 구조로 시간적 동적 변화를 충분히 반영하지 못함

Transformer: 장기 의존성 문제를 해결할 잠재력이 있지만, 긴 시퀀스 처리 시 $O(L^2)$ 연산 복잡도로 효율성 문제가 발생

Informer의 핵심 기술

Informer는 세 가지 핵심 기법을 도입하여 한계를 극복

ProbSparse Self-Attention

모든 query-key 쌍이 동일하게 중요하지 않다는 통찰에서 출발. 가장 중요한 query만 선별해 attention을 계산

Self-Attention Distilling

각 attention 레이어의 출력에 1차원 컨볼루션과 최대 풀링을 적용하여 시퀀스 길이를 절반으로 줄이면서도 핵심 패턴은 보존

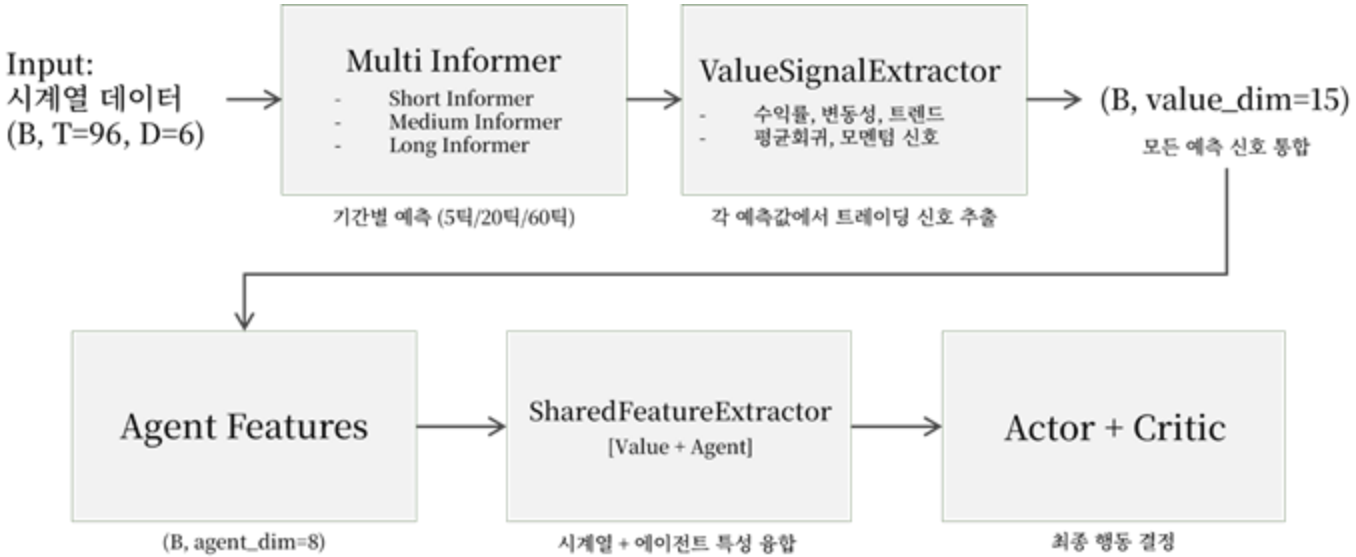
Generative Style Decoder

전체 시퀀스를 한 번에 입력받아 한 번의 forward pass로 전체 출력을 생성하여 누적 오차를 방지하고 예측 속도를 향상

이를 통해 Informer는 연산 복잡도를 $O(L^2)$에서 $O(L\log L)$로 대폭 감소시키고,
메모리 사용량을 거의 선형 수준으로 유지하며, 예측 시간을 크게 단축

Informer: 액터 크리틱 구조

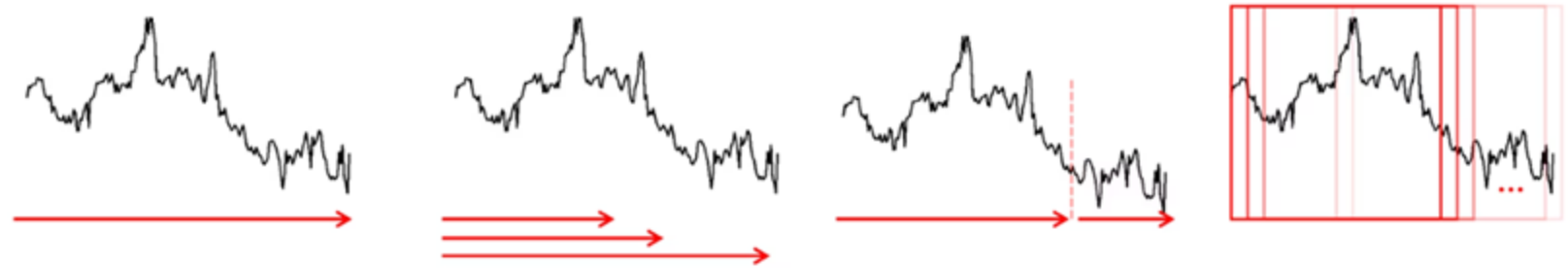
Informer의 장기 시계열 예측 능력을 강화학습 트레이딩에 활용하기 위해 MultiInformer 구조 설계



MultiInformer 특징

- 서로 다른 예측 기간(5틱, 20틱, 60틱)을 가진 세 개의 Informer 모델 병렬
- 운용단기적 시장 노이즈부터 장기적 트렌드까지 다양한 시간 척도의 정보 포착
- 동일한 96개 시점의 과거 데이터를 입력으로 사용

학습 사이클 설계



(A) Non Episodic Stream Learning	(B) Non Episodic Survival Learning	(C) Episodic Period-Based Learning	(D) Episodic Random Sampling Learning
에피소드 길이 제한 없음. 일정 스텝마다 멈춰서 신경망 업데이트. 에피소드 종료 시 이어서 학습.	에피소드 길이 제한 없음. 파산 시 전체 학습 데이터의 시작점으로 돌아가 다시 학습.	에피소드 길이 제한 있음. 에피소드 종료 시 이어서 학습 ✅ **최종 선택된 방법.**	에피소드 길이 제한 있음. 환경 리셋 시 새로운 에피소드 데이터를 가져와 시간 종속성 제거.

Non Episodic 방법론(A,B)은 자산 State의 분산이 커지는 특성으로 인해 파산을 피하면서 자산 운용을 배우기 어려움.
Episodic 방법론 중 (D)는 과적합 문제와 소극적 투자 전략으로 수렴하는 문제가 있어,
최종적으로 **Episodic Period-Based Learning(C)**을 학습에 이용함.

손실 함수 구성

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

1. 정책 업데이트

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

특정 상태에 특정 행동에서 얻은 이득을 최대화하고자 한다.
본 항을 최대화함으로써 에이전트는 정책을 학습한다.

2. 가치 업데이트

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

추정 가치 V와 실제 보상 R(GAE)을 가깝게 만든다.
본 항을 최소화함으로써 더 질 좋은 가치를 얻는다.

PPO의 기본 손실 함수에 진입 방향 정규화를 더했다. 트렌드 점수를 기준으로 규제의 정도와 타겟 분포를 제어해 장을 반영한 탐험을 유도한다.

3. 탐험 증진

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

엔트로피가 높을수록 정책은 랜덤성을 띈다.
본 항을 최대화함으로써 에이전트는 탐험을 할 수 있다.

4.진입 방향 정규화

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

$\pi_{entry\_target} = (1 - p_{mix}, p_{mix})$: 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

KL div를 이용해 **진입 방향에서의 탐험을 증진**한다.
상태별 트렌드 점수를 이용해 규제의 정도와 타겟 함수를 제어한다.

결과 분석

1. 학습-테스트 데이터 구성

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 –	2017.06.12 –

다양한 시기에 robust하게 대응하는 트레이딩 에이전트를 개발하기 위하여,
학습 구간과 테스트 구간을 다음과 같이 설정했다.

총 **15개의 구간**으로 전체 데이터를 분리한 후,
9:1 비율로 학습과 테스트 구간을 나눴다.

결과 분석

2. 결과 지표 시각화

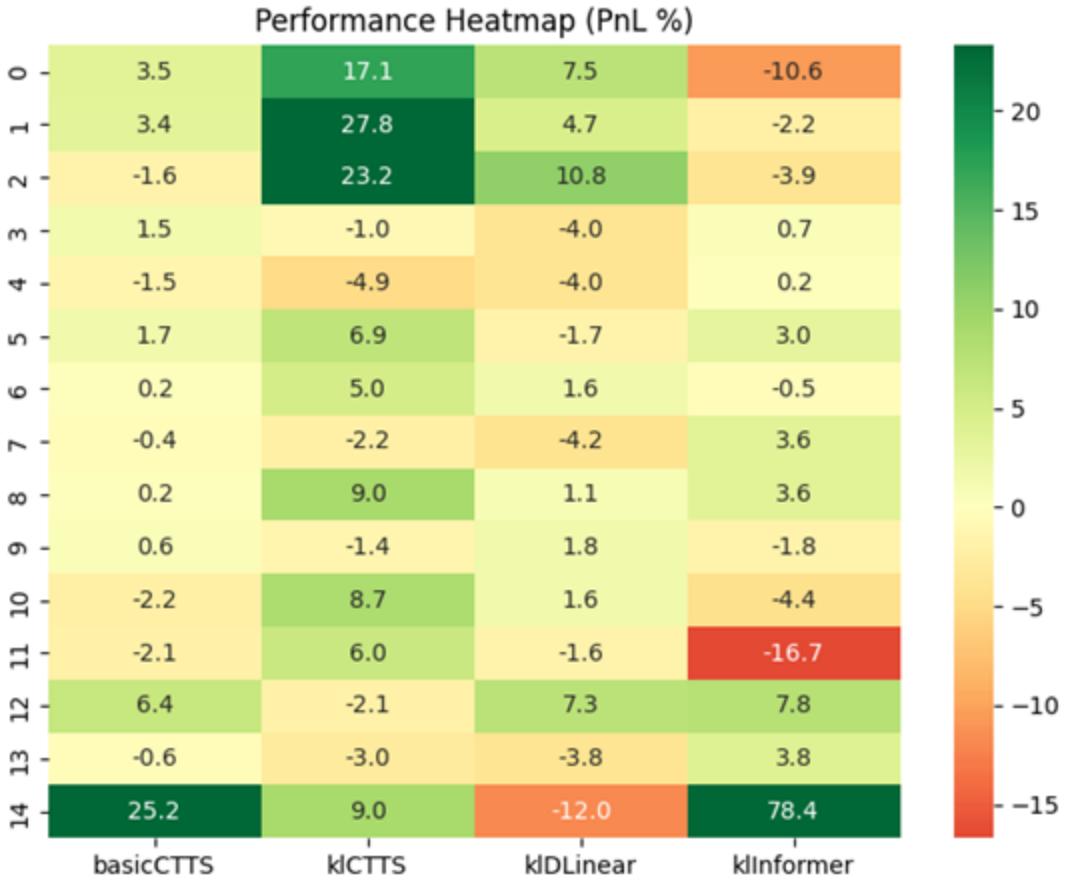
누적 PnL %

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

결과 분석

2. 결과 지표 시각화

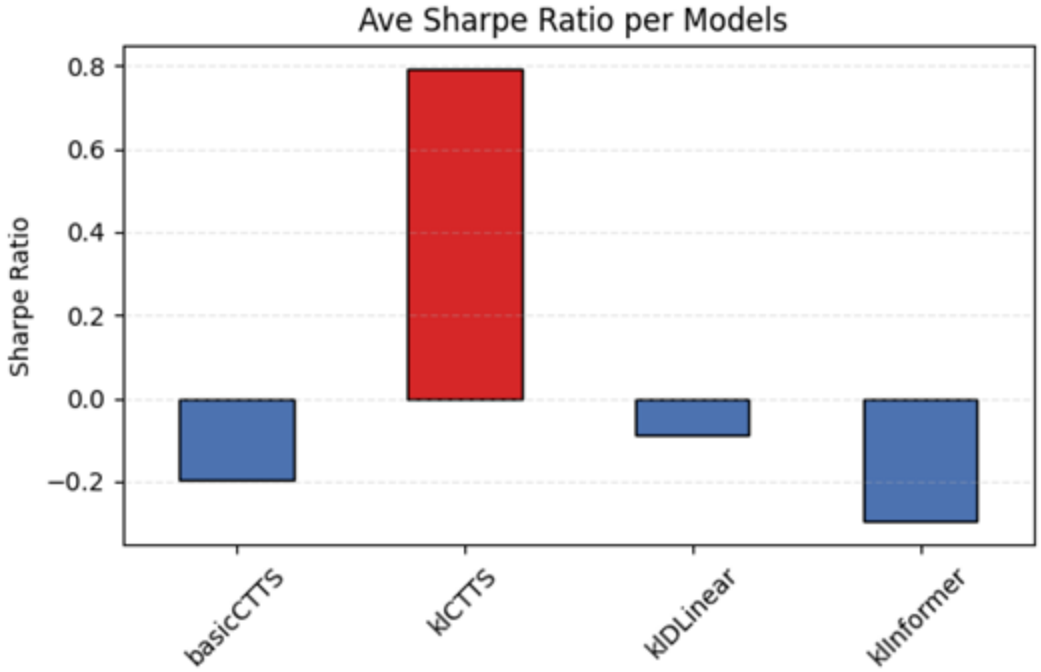
누적 PnL %



결과 분석

2. 결과 지표 시각화

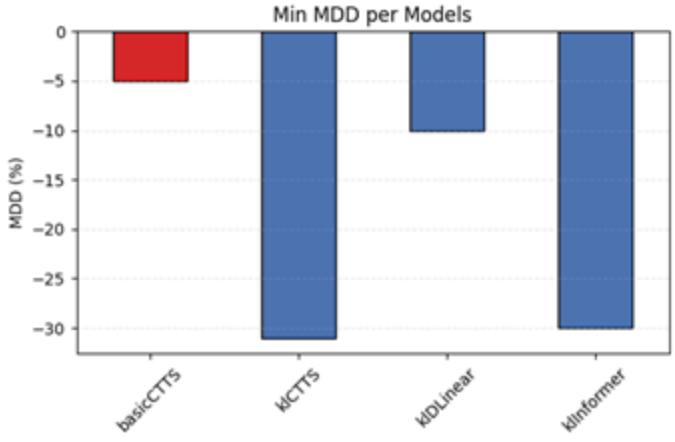
Sharpe Ratio

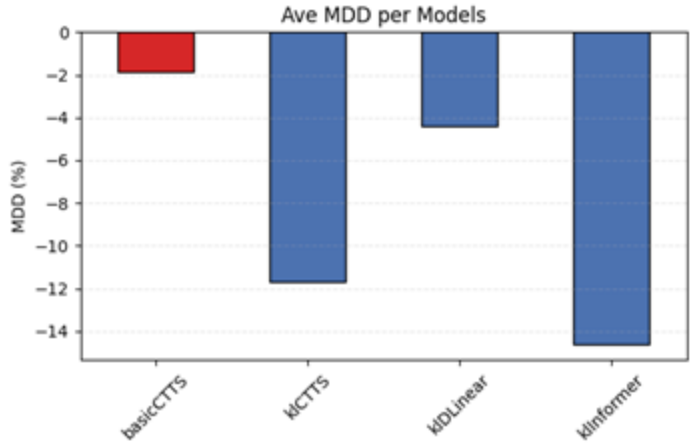


결과 분석

2. 결과 지표 시각화

MDD

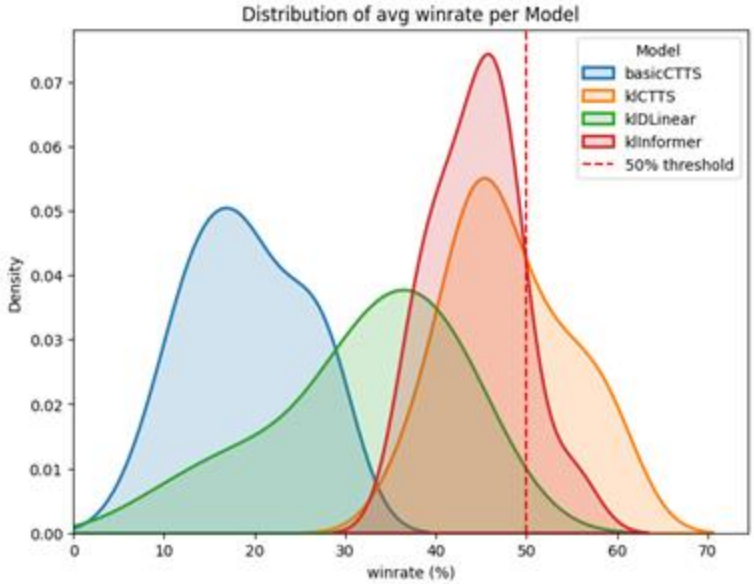


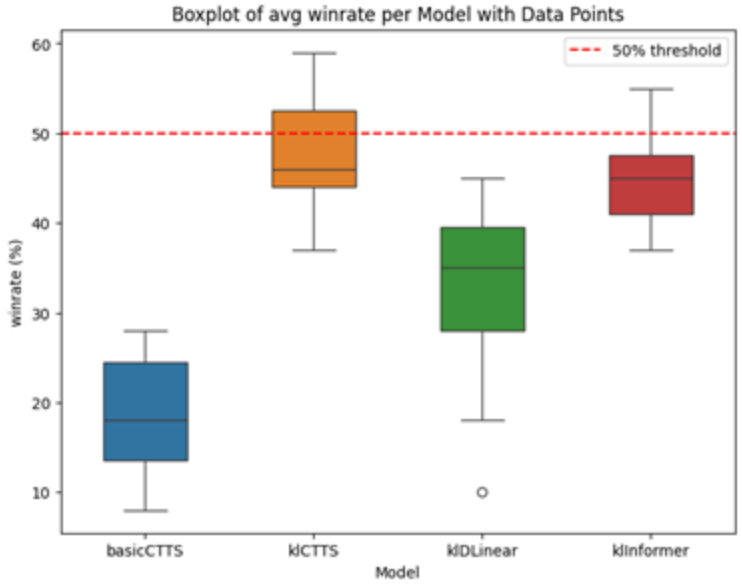


결과 분석

2. 결과 지표 시각화

승률 (%)





결과 분석

2. 결과 지표 시각화

종합

👑 **KL진입 방향 규제를 추가한 CTTS 기반 모델**

6.5 %	0.8	-12 %	48 %
평균 수익률	Sharpe Ratio	MDD	승률

수익성과 안정성을 균형 있게 확보한 투자 전략으로서의 잠재력을 가지고 있으며, 실제 투자 환경에 적용될 수 있는 중요한 가치를 지님.
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.

결과 분석

3. 후속 연구 방향

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.

감사합니다.