Deep Reinforcement Learning for Time Series: Playing Idealized Trading Games

시계열 데이터를 이용한 강화학습 기반 이상화된 트레이딩 게임

배경

- 시계열 데이터는 금융, 의료 산업 센서 등에서 많이 사용됨
- 단순 예측보다 중요한 문제는 어떤 행동을 선택할 것인가
- 주식 가격에서는 현재 행동의 결과가 미래 가격에 따라 달라짐

이전 연구 소개

- Deep Q-Learning은 Atari 게임에서 좋은 성능을 보임
- RNN, LSTM, GRU는 시계열 처리에 자주 사용됨
- CNN도 시계열 패턴 추출에 활용 가능
- 해당 논문은 GRU, LSTM, CNN, MLP를 trading game에서 비교

연구 목표

- 시계열 입력을 보고 강화학습 에이전트가 수익 전략을 학습할 수 있는지 확인
- 단일 가격 시계열에서 흐름을 학습할 수 있는지 확인
- 가격과 선행 신호 사이의 숨겨진 관계를 활용할 수 있는지 확인

Trading Game

- agent: 거래를 수행하는 인공지능
- state, s□ : 최근 40개 시점의 시계열 정보
- action, a□ : CASH, BUY, HOLD
- reward, r□ : 행동 후 얻는 수익 또는 손실
- episode : one trading game with T = 180 time step

데이터와 입력 구조

- Univariate Game
  - 입력: 가격 시계열 1가
  - 파동형 가격 데이터
  - 목적: 가격 흐름 자체를 학습하는지 확인

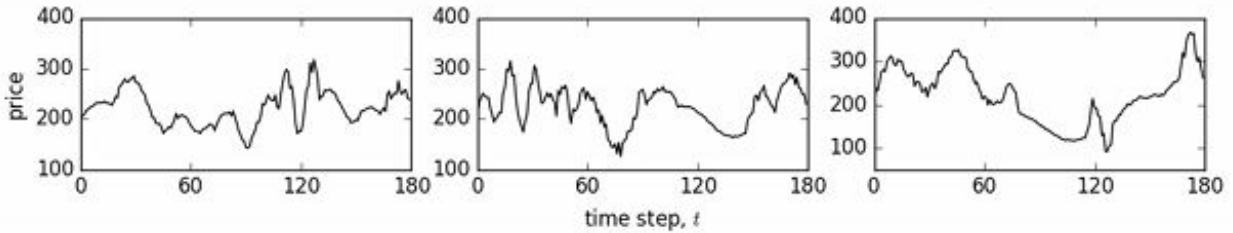


Figure 1 Examples of the Univariate game input

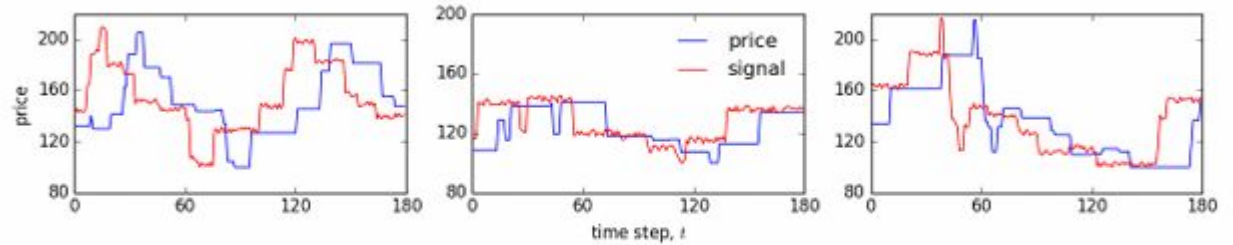


Figure 2 Examples of the Bivariate game input

- Bivariate Game
  - 입력: 가격 + Signal
  - signal은 미래 가격 변화에 대한 힌트
  - 목적: 입력 간 숨겨진 관계를 학습하는지 확인

행동, 보상 설계

- 행동
  - CASH: 주식을 보유하지 않음
  - BUY: 주식 1주 매수
  - HOLD: 보유 중인 주식 유지

- 보상
  - CASH: 0
  - BUY: 다음 가격 - 현재 가격 - 매수 비용
  - HOLD: 다음 가격 - 현재 가격
  - 매수 비용 c = 3.3

강화학습 파이프라인

1. 최근 40개 시점의 시계열 입력
2. 모델이 각 행동의 Q값 예측
3. Q값이 가장 높은 행동 선택
4. 행동 결과로 reward 획득
5. 경험 저장
6. 저장된 경험을 다시 학습에 사용

Deep Q-Learning

- Q: 현재 상태에서 특정 행동을 했을 때 기대되는 미래 보상
- Deep Q-Learning: Q 값을 신경망으로 예측
- Experience Replay: 과거 경험을 저장하고 무작위로 다시 학습
- ε-greedy: 가끔은 랜덤 행동, 대부분 Q값이 큰 행동을 선택

For episode = 1 to N
  For time step t = 1 to T
    With probability ϵ select a random valid action a_t
    Otherwise select the valid action a_t that maximize predicted Q
    Given a_t, emulator returns reward r_t and new state s_{t+1}
    Store (s_t, a_t, r_t, s_{t+1}) in memory
    For memory (s_j, a_j, r_j, s_{j+1}) in sampled minibatch
      perform gradient descent on $Q(s_j, a_j)$ using target value

$$Q_{\text{target}}(s_j, a_j) = \begin{cases} r_j & \text{game end at } j \\ r_j + \gamma \max\limits_{\text{valid } a} Q(s_{j+1}, a) & \text{otherwise} \end{cases}$$

    end for
  end for
end for

모델 구조

- MLP: 기본 신경망
- CNN: 짧은 구간의 국소 패턴 추출
- GRU: 시간 흐름을 기억하는 순환 신경망
- LSTM: 장기 의존성을 다루는 순환 신경망

모든 모델 출력: CASH, BUY, HOLD의 Q값 3

학습 설정과 평가 지표

- 학습: 1000 episodes
- 테스트: 1000 episodes
- 한 episode 길이: 180 time steps
- discount factor $\gamma = 0.8$

- 평가지표
  - Mean P&L
  - P&L > 0 비율

Univariate Game

- 가격 시계열 하나만 보고 판단하는 게임
- GRU 기반 모델이 가장 좋은 성능을 보임

Table 2 Test results for the profit and loss (P&L) generated per episode

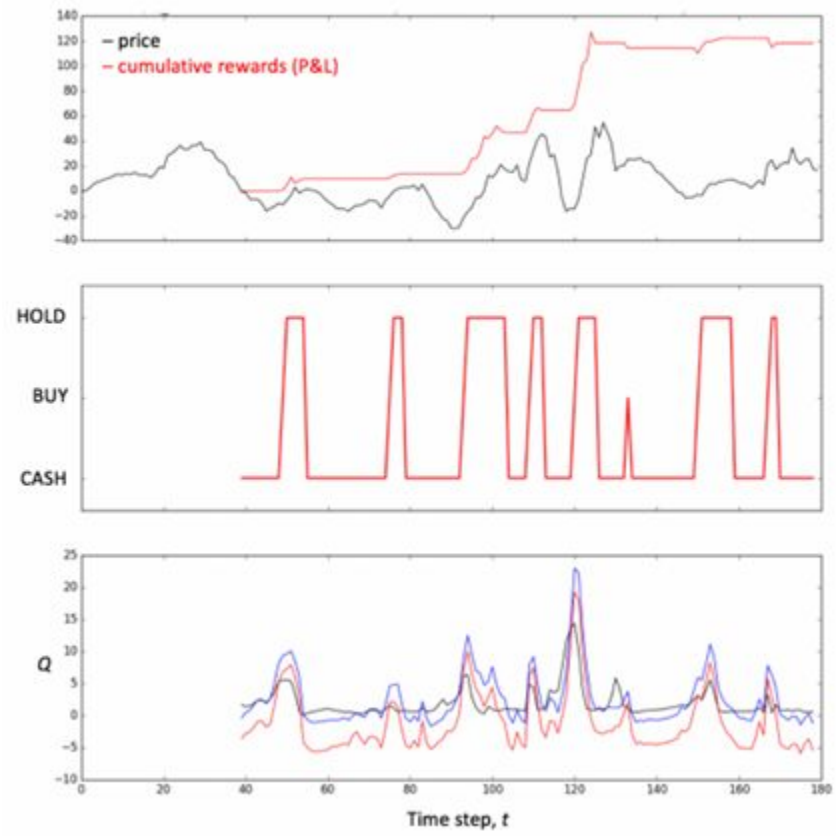
	Univariate game					Bivariate game				
		In-sample		Out-of-sample			In-sample		Out-of-sample	
	#param	Mean P&L	P&L> 0	Mean P&L	P&L> 0	#param	Mean P&L	P&L>0	Mean P&L	P&L>0
MLP 16x4	1,523	104.3	100%	96.9	97%	2,163	49.4	92%	40.5	96%
MLP 16x5	1,795	103.2	100%	91.7	100%	2,435	47.4	94%	43.8	94%
MLP-32x4	4,579	122.4	100%	86.1	96%	5,859	49.3	92%	37.4	95%
MLP-32x5	5,635	125.0	100%	89.8	99%	6,915	52.7	95%	42.0	96%
GRU-8x3	1,227	116.1	100%	115.6	100%	1,251	35.0	81%	31.8	86%
GRU-16x3	4,627	114.0	100%	112.1	100%	4,675	44.0	84%	35.0	91%
GRU-16x2	3,043	120.7	99%	115.2	100%	3,091	46.5	91%	31.1	88%
GRU-32x3	17,955	143.3	100%	116.1	99%	18,051	51.5	94%	37.2	94%
LSTM-8x3	1,579	58.9	90%	68.0	94%	1,611	37.6	80%	32.5	84%
LSTM-16x3	5,971	84.8	98%	79.4	98%	6,035	44.5	91%	38.4	94%
LSTM-16x2	3,859	106.3	100%	107.8	100%	3,923	44.0	87%	37.8	90%
LSTM-32x3	23,203	134.1	100%	98.5	97%	23,331	44.3	90%	31.0	89%
CNN-8x3	2,883	89.0	98%	77.3	94%	2,907	39.1	86%	34.7	87%
CNN-16x3	5,235	106.0	98%	86.7	95%	5,283	38.8	85%	28.5	85%
CNN-16x2	8,291	108.5	100%	77.1	95%	8,339	46.4	91%	33.8	89%
CNN-32x3	12,243	120	100%	80.8	96%	12,339	32.3	85%	15.4	72%

Brivariate Game

- 가격 + signal을 보고 판단하는 게임
- MLP 기반 모델이 가장 좋은 성능을 보임
- signal이 직접적인 힌트 역할을 했기 때문에 단순 모델도 좋은 성능

Table 2 Test results for the profit and loss (P&L) generated per episode

	Univariate game					Bivariate game				
		In-sample		Out-of-sample			In-sample		Out-of-sample	
	#param	Mean P&L	P&L> 0	Mean P&L	P&L> 0	#param	Mean P&L	P&L>0	Mean P&L	P&L>0
MLP 16x4	1,523	104.3	100%	96.9	97%	2,163	49.4	92%	40.5	96%
MLP 16x5	1,795	103.2	100%	91.7	100%	2,435	47.4	94%	43.8	94%
MLP-32x4	4,579	122.4	100%	86.1	96%	5,859	49.3	92%	37.4	95%
MLP-32x5	5,635	125.0	100%	89.8	99%	6,915	52.7	95%	42.0	96%
GRU-8x3	1,227	116.1	100%	115.6	100%	1,251	35.0	81%	31.8	86%
GRU-16x3	4,627	114.0	100%	112.1	100%	4,675	44.0	84%	35.0	91%
GRU-16x2	3,043	120.7	99%	115.2	100%	3,091	46.5	91%	31.1	88%
GRU-32x3	17,955	143.3	100%	116.1	99%	18,051	51.5	94%	37.2	94%
LSTM-8x3	1,579	58.9	90%	68.0	94%	1,611	37.6	80%	32.5	84%
LSTM-16x3	5,971	84.8	98%	79.4	98%	6,035	44.5	91%	38.4	94%
LSTM-16x2	3,859	106.3	100%	107.8	100%	3,923	44.0	87%	37.8	90%
LSTM-32x3	23,203	134.1	100%	98.5	97%	23,331	44.3	90%	31.0	89%
CNN-8x3	2,883	89.0	98%	77.3	94%	2,907	39.1	86%	34.7	87%
CNN-16x3	5,235	106.0	98%	86.7	95%	5,283	38.8	85%	28.5	85%
CNN-16x2	8,291	108.5	100%	77.1	95%	8,339	46.4	91%	33.8	89%
CNN-32x3	12,243	120	100%	80.8	96%	12,339	32.3	85%	15.4	72%


price
cumulative rewards (P&L)
HOLD
BUY
CASH
Q
Time step, t

한계와 향후 연구

- 실제 시장 데이터가 아니라 인공 데이터 사용
- 거래 비용, 유동성, 슬리피지, 리스크 관리 등이 단순화됨
- 실제 투자 성능으로 바로 해석하면 안 됨
- 향후 더 현실적인 trading game과 고급 강화학습 기번 필요