

동적 자산배분 toy 모델 구축

강화학습을 활용한 동적 자산배분

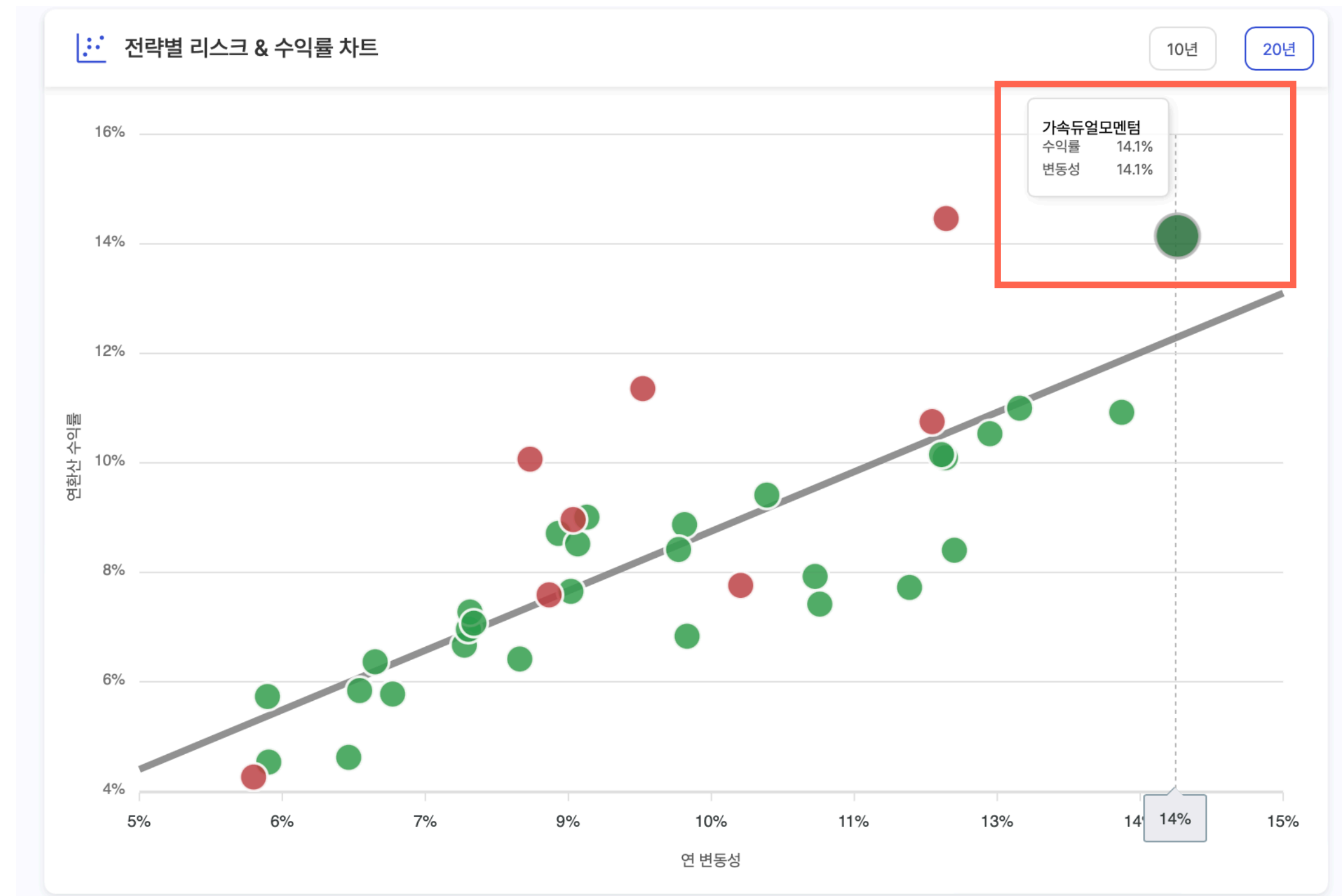
윤경호

목차

- 개요
- 자산군 데이터 분석
- 모델
- 결과
- Future work

개요

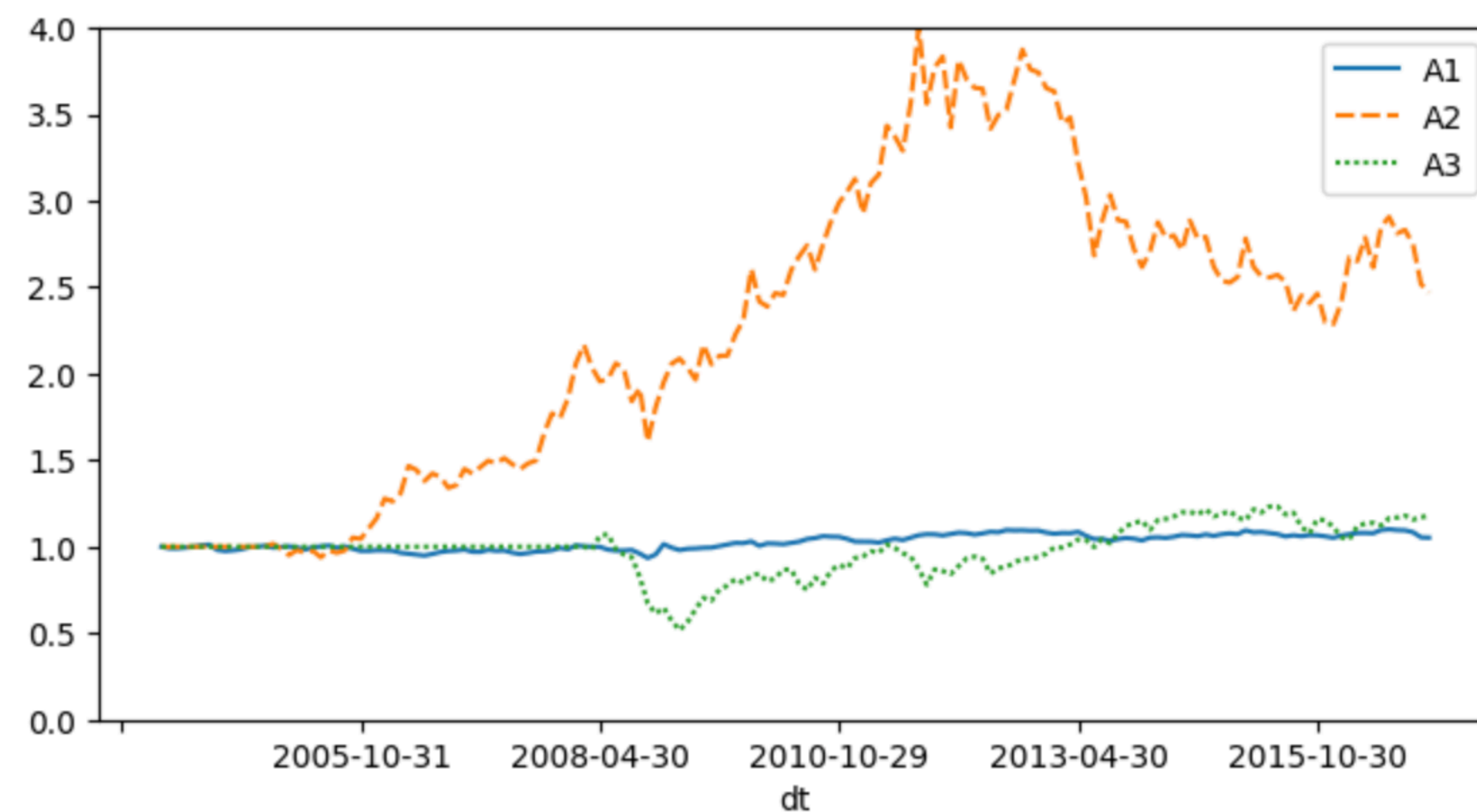
- 모멘텀을 활용한 룰베이스 기반 전략을 강화학습으로 재설계
 - 가속 듀얼 모멘텀 전략 방향성 참고
- 모멘텀 기반 룰베이스 전략, Efficient Frontier의 Sharpe maximized 전략과 비교



자산 배분 알고리즘 별 리스크 & 수익률 (출처: snowball72.com)

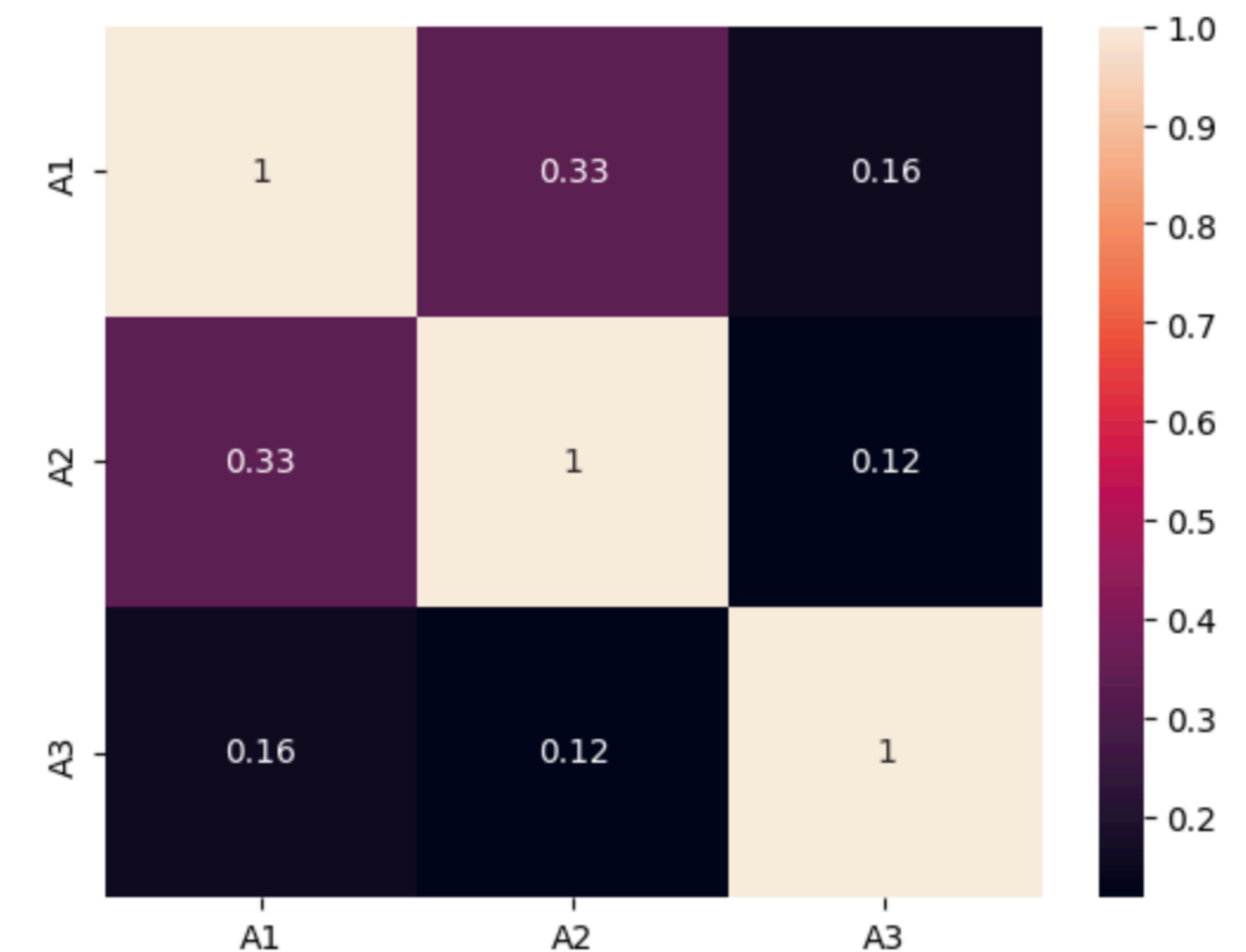
자산군 데이터 분석

- 전체 데이터 기간
 - 2003.09.22~2023.07.20 (5174영업일)
 - ~ 2004.11.18의 A2, A3는 데이터 변동이 없음
- 학습 / 테스트
 - 학습 기간: ~ 2016.12.31
 - 테스트 기간: 2017.01.01 ~ 2023.07.20 (1709영업일)



학습 기간 자산군 별 가격 변화

- 학습 기간 중, 자산군 별 cagr 차이가 큼
 - Cagr: A2(8.4%) > A3(0.9%) > A1(0.5%)
- A2 자산군의 경우, Sharpe, cagr 모두 두 자산군 대비 높음
 - Sharpe: A2(0.54) > A1(0.23) > A3(0.14)
- 자산군간 correlation이 낮아, A2를 중심으로 A1, A3를 조합하여 수익률 감소를 최소화하면서 sharpe를 증가 시키는 mp 전략이 유효할 것으로 판단



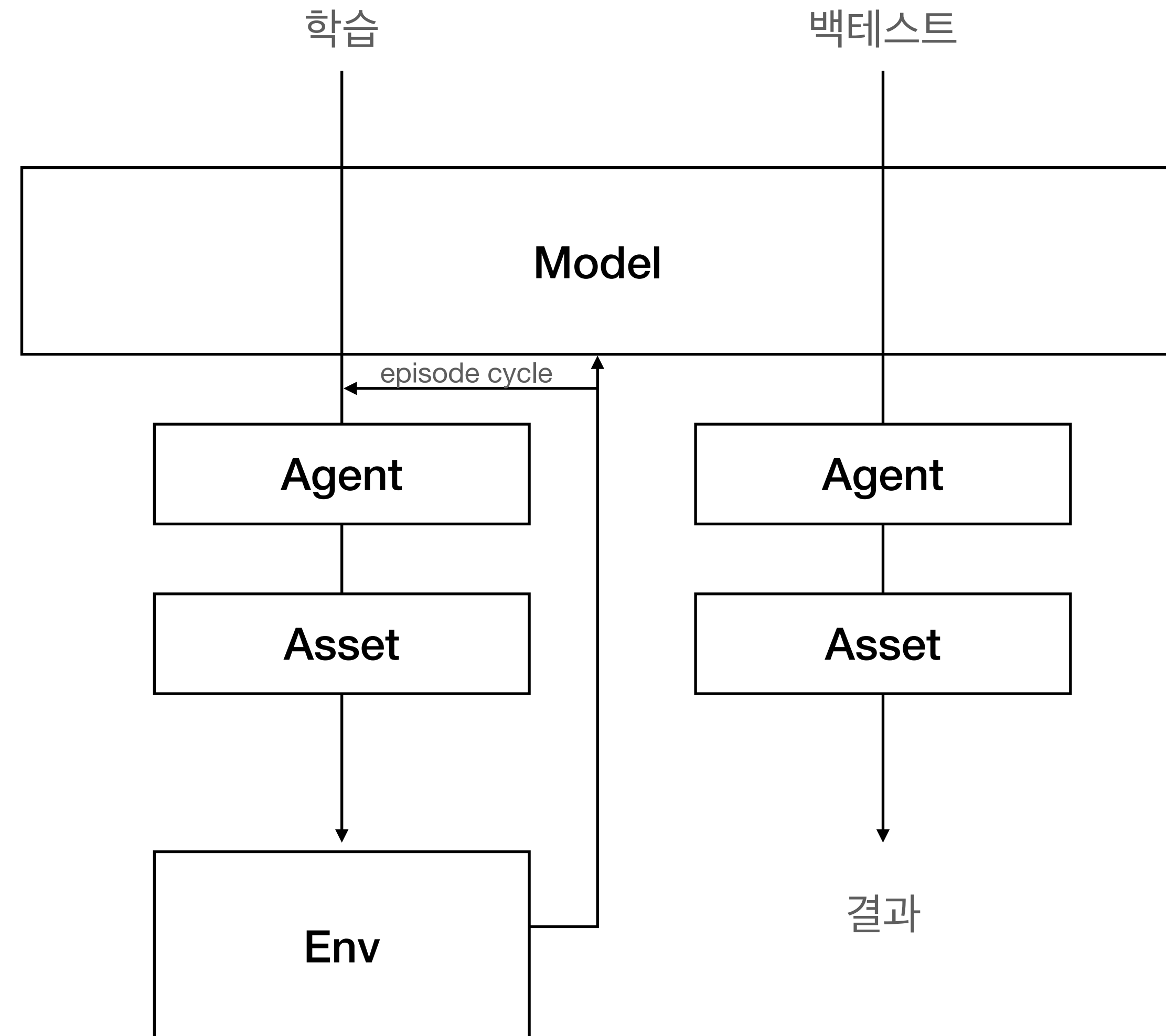
학습 기간 자산군 별 correlation

모델 - 구조

- 자산군 별 momentum score를 받아 Sharpe의 max를 추구하는 continuous한 mp를 찾는 RL 모델
- 구조
 - State
 - [3*3] (자산군 별 1,3,6개월 momentum score)
 - Action
 - [3] (자산군 별 비중)
 - 자산군 별 $0 \leq x \leq 1$ continuous value 예측, 예측 결과에 softmax 적용
 - Reward
 - 학습 종료 시 sharpe ratio
- 학습 및 백테스트 방식
 - 월 말 1회 리밸런싱 진행, 매 월 리밸런싱 진행 직전 momentum score를 받아 mp 생성
 - A2 자산군 기본 비중=50%, 모델이 예측한 mp를 바탕으로 A1, A2, A3의 나머지 비중을 채움
 - 매매 비용은 매수/매도 10bp 가정
 - 소수점 매매 불가 가정 (buffer 금액 < 2% 내외 존재 가능)
 - risk_free_profit = 0 가정

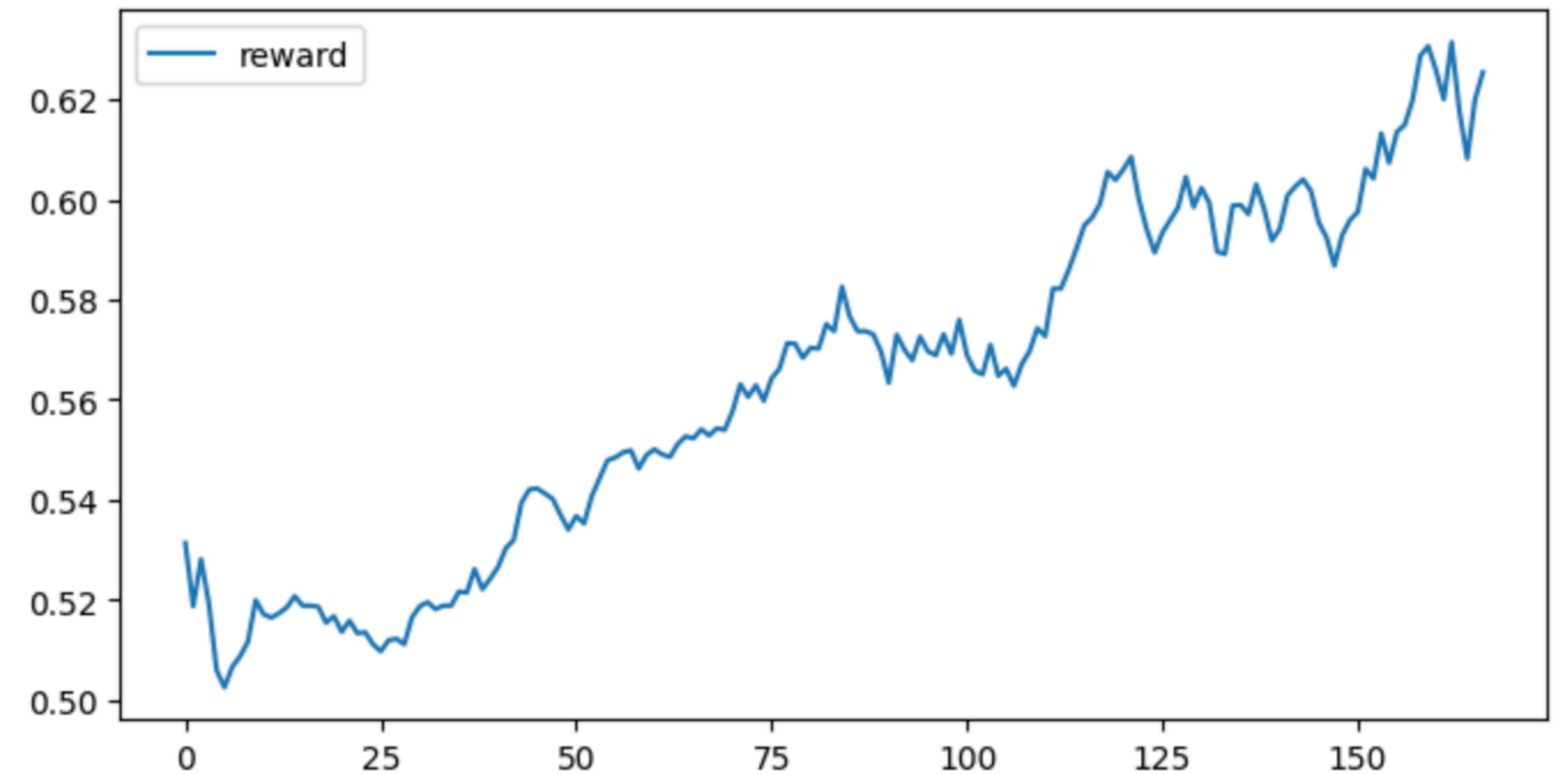
모델 - 프로젝트 구조

- Project
 - model.py
 - 데이터 입력, 학습 실행, 백테스트 실행 wrapper
 - utils.py
 - Agent
 - 백테스트, 학습하는 가상 계좌 클래스
 - Asset
 - 자산군 별 Ratio 클래스
 - env.py
 - 학습 시 활용하는 environment 클래스
 - model/model.zip
 - 학습 모델 파일
 - visualization.ipynb
 - 데이터 분석 스크립트



결과

- 학습
 - 학습 환경
 - Mac M1 2020 (using cpu)
 - torch, stable-baselines 활용
 - 소요시간: 약 3시간
 - 학습 중단 조건
 - 1 epoch = 50000 step
 - 100~200 epoch 중 max reward 모델 선택
 - learning_rate 외 하이퍼 파라미터 튜닝 X / 데이터 엔지니어링 X
 - 성능을 높이는 것이 아닌, 컨셉 확인 목적 모델



epoch reward 변화

결과

- 테스트 기간: 2017.01.01 ~ 2023.7.20 (리밸런싱 79회)
- 비교군
 - 룰베이스 모멘텀 전략
 - 1개월, 3개월, 6개월 모멘텀 sum이 최대인 자산군 매수
 - 월 주기 리밸런싱
 - Efficient frontier 전략
 - Ef선에서 Sharpe max하는 mp로 리밸런싱
 - 월 주기 리밸런싱
- 결과 분석
 - RL 모델을 활용했을 때 룰베이스로 모멘텀을 활용한 동적 자산배분 전략 보다 높은 sharpe, 낮은 cagr 기록
 - 모멘텀을 활용해 max 자산군만 매수하는 전략보다, 다른 자산군을 섞는 전략이 sharpe 수치 상 더 좋은 성과를 보일 수 있음
 - Reward 설계를 sharpe를 max하는 것으로 하였기 때문에, cagr에서는 약간 부족한 결과를 보임
 - Efficient Frontier를 활용한 정적 자산 배분 전략 보다 cagr에서는 크게 앞섰지만, sharpe에서는 큰 차이를 내지 못함
 - MDD는 두 전략 대비 높은 모습을 보임
 - 짧은 기간에서의 데이터 하락에서 손해가 큰 케이스 존재

Algorithm	CAGR	Sharpe	MDD
룰베이스 모멘텀 전략	5.87%	0.41	-17.42%
Efficient Frontier 전략	2.47%	0.41	-14.32%
RL 모델	5.65%	0.51	-24.42%

Future work

- 현재 학습한 모델은 위험 자산(A2) 비중 50%를 기본으로, 모델이 예측한 값을 A1, A2, A3 자산군에 채우는 모델
 - 학습 인프라가 충분하다면 서치 스페이스를 증가시켜 100% 모두 학습을 통해 예측하는 접근 필요
- 학습 시 Sharpe를 maximize하는 모델 외에 mdd에 대한 피드백도 reward에 추가하여 mdd 측면에서도 개선된 모델 개발
 - Sharpe와 mdd의 reward scaling 방안에 대한 고민이 필요
- Poc 수준의 toy 프로젝트였기 때문에, 테스트 방식이나 학습 방식에도 부족한 점이 많았음
 - 학습 시 고정된 시작, 종료 시점이 아닌 랜덤한 시점에서 시작과 종료하여 더욱 많은 케이스에 대한 학습 필요
 - 테스트 셋의 다양화 필요 (조금 더 다양한 시장 국면에 대한 테스트)
- state와 그에 대한 action 데이터 분석을 통해 학습에 활용한 모멘텀 점수와 모델이 예측한 값의 상관관계 분석 필요
 - “특정 모멘텀 수치를 가졌을 때, 어떠한 mp를 갖는 것이 기대값 상 유리하다” 와 같은 논리 필요
- 하이퍼 파라미터 튜닝, state 데이터 엔지니어링을 통해 학습 수렴 속도, 성능적 개선 필요