SAC-AP:Soft Actor Critic based Deep Reinforcement Learning for Alert Prioritization

(경보 우선순위 지정을 위한 소프트 액터-크리틱 기반 심층 강화학습)

발표자: 권우현

팀원: 권우현, 정민성, 이정섭

2025.09.05

목 차

논문 소개

연구 배경 및 문제 정의

문제 정의

- 침입 탐지 시스템은 대량의 경보 생성, 그 중 대부분이 오탐
- 보안 전문가의 위협 식별 및 조사를 위해 경보 우선순위 지정 필요성 증대

기존 경보 우선순위 지정 방식의 한계

- 지도/비지도 머신러닝: 대규모 데이터셋에서 성능 저하, 유동적인 공격 패턴 대응 불가
- DDPG: 과적합 문제, 탐색 능력 부족 및 무작위 환경에 부적합

해결방안

- 유동적 공격 패턴 대응
- 대규모 데이터셋에서도 효과적
- 과적합 해결 및 효율적인 탐색 능력 보유
- 무작위 환경에서도 안정적인 성능

배경 지식

강화학습(RL) 기본 개념

강화학습은 지능형 에이전트가 환경과 상호작용하는 과정

- 에이전트는 상태(S_t)를 관찰하고, 정책 $\pi(a|s)$에 따라 행동(a_t)
- 결과로 보상(R_t)을 받고 다음 상태(S_{t+1})로 이동
- 목표는 총 보상을 최대화하고 최적 정책을 얻는 것

핵심 함수

가치 함수(Value Function)

$$V_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|s_t = s\right]$$

행동 가치 함수(Action Value Function)

$$Q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}|s_t = s, a_t = a\right]$$

액터-크리틱 방법론 개념

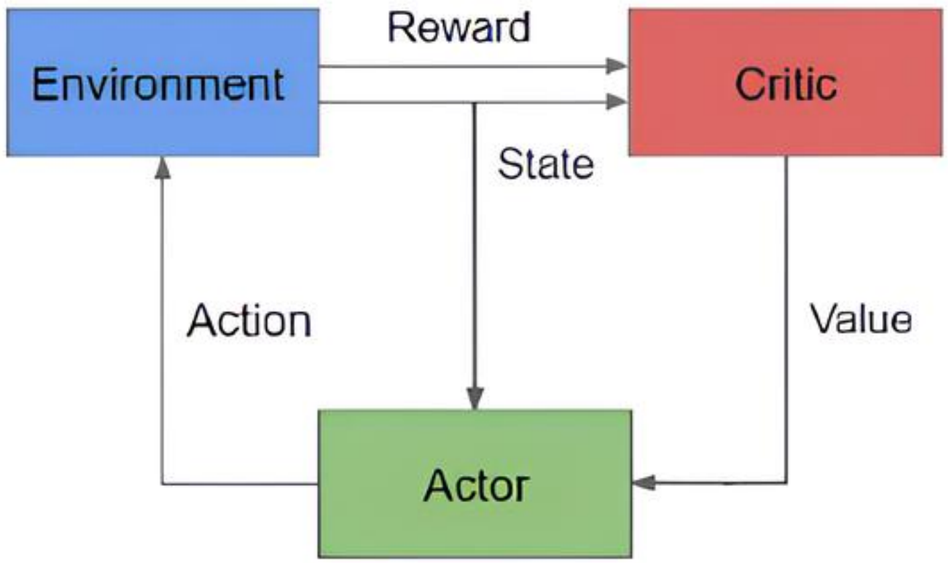


Fig1. 환경, 액터, 크리틱 상호작용 흐름도

환경

에이전트가 상호작용하는 환경으로 모든 정책 및 상태 포함

액터로부터 행동을 입력 받고, 보상과 다음 상태를 에이전트에게 제공

액터

정책에 따라서, 현재 상태에서의 행동 결정

환경으로부터 입력 받은 상태 기반으로 행동을 선택해 환경에 전달

크리틱이 제공하는 가치 정보를 통해 정책 업데이트

크리틱

액터의 행동에 대한 가치 평가 담당

환경으로부터 보상과 상태를 입력 받아 액터의 현 상태에서 얻을 수 있는 총 기대가치 예측

소프트 액터-크리틱(SAC) 알고리즘

SAC
최대 엔트로피 강화학습 프레임워크 기반
Off-policy 액터-크리틱 알고리즘
기대보상을 최대화하는 동시에 엔트로피 최대화

목표 함수

$$\pi^*(.|s_t) = argmax_\pi E_\pi[\sum_t R_t + \beta H(\pi(.|s\_t))]$$

표현식 설명

$\pi^*(.|s_t)$: 상태 s_t에서의 엔트로피와 그에 따른 기대보상을 최대화하는 정책의 확률 분포
R_t: 시점 t에서의 보상
β: 최적 정책에 영향을 미치는 상관계수로 0이면 결정적, 아니라면 확률적
$H(\pi(.|s_t))$: 정책 π의 통제 하에 있는 방문했던 상태의 정책에 대한 엔트로피
$argmax_\pi E_\pi[..]$: 정책이 π일 때, 괄호 안 표현식의 값이 가질 수 있는 평균이 최대값이 되는 정책 탐색

SAC–AP

SAC-AP 시스템 아키텍처

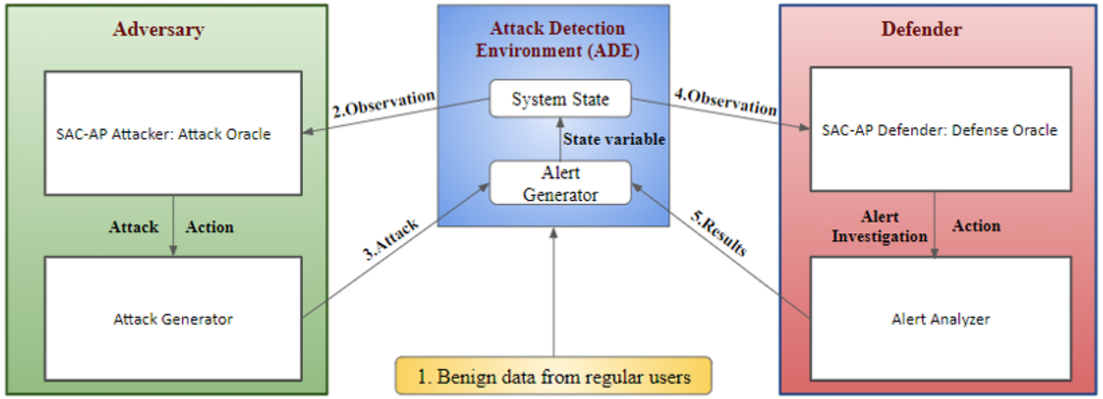


Fig. 2. System Architecture

일반 사용자

시스템의 등록된 사용자로, 정상적인 활동을 하지만 경우에 따라 이들의 활동이 오탐(False Positive) 경보를 발생시킬 수 있음

공격 탐지 환경(ADE)

공격자의 행동에 따라 경보를 생성하는 환경으로, 시스템 상태와 경보 생성기로 구성됨, 정상 사용자의 활동에 의해서도 경보가 발생 가능

공격자

시스템의 비인가 사용자로, 공격 오라클과 공격 생성기로 구성됨, 공격자는 시스템 상태와 방어자의 이전 전략을 알고 있으며, 공격 횟수는 예산(D)에 따라 정해짐

방어자

방어 오라클과 경보 분석기로 구성됨. 방어자는 예산(B) 제약 내에서 경보를 조사하며, 공격으로 인한 경보를 조사하지 못할 경우 공격자가 성공함

SAC-AP 알고리즘(1)

Algorithm 1 SAC-AP Algorithm: Compute the pure-strategy best response of player v when its opponent takes mixed strategy σ_{-v}.

Input: The set of opponent's pure strategies, Π_{-v} and mixed strategy of the opponent, σ_{-v};
Output: The policy network of player v, $\pi_v(O_v|\theta_v^{\pi})$, the value network of player v, $V(O_v|\theta_v^{v})$ and the critic networks of player v, $Q_{1,2}(O_v, \alpha_v|\theta_v^{Q_{1,2}})$;

Randomly initialize $\pi_v(O_v|\theta_v^{\pi})$, $V(O_v|\theta_v^{v})$ and $Q_{1,2}(O_v, \alpha_v|\theta_v^{Q_{1,2}})$;
Initialize replay memory D;
for $episode = 0, M-1$ **do**
  Initialize the system state $< N^{(0)}, M^{(0)}, S^{(0)} >$;
  Sample opponent's policy π_{-v}, with its mixed strategy σ_{-v} over Π_{-v};
  for k=0, k-1 **do**
    With probability ϵ select random action $\alpha_v^{(k)}$;
    Otherwise select $\alpha_v^{(k)} = \pi_v(O_v|\theta_v^{\pi})$;
    Execute $\alpha_v^{(k)}$ and $\alpha_{-v}^{(k)} = \pi_{-v}(O_{-v}^{(k)})$, observe reward r_v^k and transit the system state to s^{k+1};
    Store transition $< O_v^k, \alpha_v^k, r_v^k, O_v^{k+1} >$ in D;
    Sample a random minibatch of N transitions $< O_v^k, \alpha_v^k, r_v^k, O_v^{k+1} >$ from D;
    Set $Q = min\big[E\big(Q_1(O_v^k, \pi_v(\theta_v^{\pi})\big), E\big(Q_2(O_v^k, \pi_v(\theta_v^{\pi})\big)\big]$;
    Set $J_{\pi} = E\big[-Q + \beta log\big(\pi_v(O_v|\theta_v^{\pi})\big)\big]$;
    Set $V' = E\big[Q - log\big(\pi_v(\theta_v^{\pi})\big)\big]$;
    Set $J_v = E\big[\frac{1}{2} * \big(V' - V(\theta_v^{v})\big)^2\big]$;
    Set $Q' = r_v^k + \gamma * \pi_v(\theta_v^{'})$;
    Set $J_{Q_1} = \frac{1}{2} * E\big[\big(Q' - Q_1(\theta_v^{Q_1})\big)\big]^2$;
    Set $J_{Q_2} = \frac{1}{2} * E\big[\big(Q' - Q_2(\theta_v^{Q_2})\big)\big]^2$;
    $\theta_v^{Q_1} \leftarrow \theta_v^{Q_1} - \nabla J_{Q_1}$;
    $\theta_v^{Q_2} \leftarrow \theta_v^{Q_1} - \nabla J_{Q_2}$;
    $\theta_v^{\pi} \leftarrow \theta_v^{\pi} - \nabla J_{\pi}$;
    $\theta_v^{v} \leftarrow \theta_v^{v} - \nabla J_v$;
    $\theta_v^{v'} \leftarrow \tau\theta_v^{v} + (1-\tau)\theta_v^{v'}$;
  end for
end for
return player v's policy network $\pi_v(O_v|\theta_v^{\pi})$;

1. 신경망 및 메모리 초기화

액터 네트워크(π_v), 가치 네트워크(V), 두 개의 크리틱 네트워크($Q_{1,2}$) 무작위 초기화

리플레이 메모리 D를 생성해 경험 데이터 저장 준비

2. Episode 시작

각 episode 마다 시스템 상태 초기화

상대의 혼합 정책 σ_v로부터 하나의 결정적 정책 π_{-v}를 샘플링

이는, 에이전트가 특정 상대방 정책에 대한 최선의 응답을 학습하게끔 지원.

3. 환경 상호작용 (Time Step k)

- 에이전트의 행동 선택

$\epsilon-$ greedy 전략에 따라 확률 ϵ로 무작위 행동 $\alpha_v^{(k)}$를 선택하거나,

액터 네트워크가 $\pi(O_v|\theta_v^{\pi})$가 제안하는 행동 선택

- 행동 실행 및 보상 관찰

에이전트는 선택한 행동 $\alpha_v^{(k)}$를 실행, 상대방은 샘플링된 정책 π_{-v}에 따라 행동 $\alpha_v^{(k)}$ 를 실행

에이전트는 보상$r_v^{(k)}$를 받고 시스템 상태가 s_{k+1}로 전이

- 경험 저장

관찰 정보 $O_v^{(k)}$, 행동 $\alpha_v^{(k)}$, 보상 $r_v^{(k)}$, 다음 시간 관찰 정보 O_v^{k+1}를 리플레이 메모리 D에 저장

SAC-AP 알고리즘(2)

Algorithm 1 SAC-AP Algorithm: Compute the pure-strategy best response of player v when its opponent takes mixed strategy σ_{-v}.

Input: The set of opponent's pure strategies, Π_{-v} and mixed strategy of the opponent, σ_{-v};
Output: The policy network of player v, $\pi_v(O_v|\theta_v^{\pi})$, the value network of player v, $V(O_v|\theta_v^{v})$ and the critic networks of player v, $Q_{1,2}(O_v, \alpha_v|\theta_v^{Q_{1,2}})$;

Randomly initialize $\pi_v(O_v|\theta_v^{\pi})$, $V(O_v|\theta_v^{v})$ and $Q_{1,2}(O_v, \alpha_v|\theta_v^{Q_{1,2}})$;
Initialize replay memory D;
for $episode = 0, M-1$ **do**
  Initialize the system state $< N^{(0)}, M^{(0)}, S^{(0)} >$;
  Sample opponent's policy π_{-v}, with its mixed strategy σ_{-v} over Π_{-v};
  for k=0, k-1 **do**
    With probability ϵ select random action $\alpha_v^{(k)}$;
    Otherwise select $\alpha_v^{(k)} = \pi_v(O_v|\theta_v^{\pi})$;
    Execute $\alpha_v^{(k)}$ and $\alpha_{-v}^{(k)} = \pi_{-v}(O_{-v}^{(k)})$, observe reward r_v^k and transit the system state to s^{k+1};
    Store transition $< O_v^k, \alpha_v^k, r_v^k, O_v^{k+1} >$ in D;
    Sample a random minibatch of N transitions $< O_v^k, \alpha_v^k, r_v^k, O_v^{k+1} >$ from D;
    Set $Q = min\big[E\big(Q_1(O_v^k, \pi_v(\theta_v^{\pi})\big), E\big(Q_2(O_v^k, \pi_v(\theta_v^{\pi})\big)\big]$;
    Set $J_{\pi} = E\big[-Q + \beta log\big(\pi_v(O_v|\theta_v^{\pi})\big)\big]$;
    Set $V' = E\big[Q - log\big(\pi_v(\theta_v^{\pi})\big)\big]$;
    Set $J_v = E\big[\frac{1}{2} * \big(V' - V(\theta_v^{v})\big)^2\big]$;
    Set $Q' = r_v^k + \gamma * \pi_v(\theta_v^{v'})$;
    Set $J_{Q_1} = \frac{1}{2} * E\big[\big(Q' - Q_1(\theta_v^{Q_1})\big)\big]^2$;
    Set $J_{Q_2} = \frac{1}{2} * E\big[\big(Q' - Q_2(\theta_v^{Q_2})\big)\big]^2$;
    $\theta_v^{Q_1} \leftarrow \theta_v^{Q_1} - \nabla J_{Q_1}$;
    $\theta_v^{Q_2} \leftarrow \theta_v^{Q_1} - \nabla J_{Q_2}$;
    $\theta_v^{\pi} \leftarrow \theta_v^{\pi} - \nabla J_{\pi}$;
    $\theta_v^{v} \leftarrow \theta_v^{v} - \nabla J_v$;
    $\theta_{v'}^{v} \leftarrow \tau\theta_v^{v} + (1-\tau)\theta_v^{v'}$;
  end for
end for
return player v's policy network $\pi_v(O_v|\theta_v^{\pi})$;

4. 네트워크 업데이트

리플레이 메모리 D 에서 무작위 Minibatch를 샘플링하여 신경망을 업데이트

- Critic Networks($Q_{1,2}$) 업데이트

두개의 Q-값 중 최솟값을 활용하여 Target Q –값 Q'을 계산

Q_1, Q_2 각각의 손실함수 J_{Q_1}, J_{Q_2}를 최소화하도록 가중치 $\theta_v^{Q_1}, \theta_v^{Q_2}$를 그래디언트 감소 방식으로 업데이트

- Value Network (V) 업데이트

Soft Q-값과 정책 엔트로피를 기반으로 목표 가치 V'를 계산

손실 함수 J_v를 최소화하도록 가중치 θ_v^{π}를 그래디언트 감소 방식으로 업데이트

- 액터 네트워크 (π) 업데이트

두 크리틱 네트워크의 Q-값 중 최솟값과 정책 엔트로피 $\beta\mathrm{H}\big(\pi(.|s_t)\big)$를 사용하여 액터의 손실 함수 J_v를 정의

J_{π}를 최소화하도록 가중치 θ_v^{π} 를 그래디언트 감소 방식으로 업데이트해 보상과 엔트로피 동시 최대화

- 목표 가치 네트워크 업데이트

학습 안정성을 위해 목표 가치 네트워크의 가중치 $\theta_v^{\mathrm{v}\prime}$를 지수 이동 평균 방식으로 업데이트

5. 최적 정책 반환

정해진 수의 episode가 완료 시, 학습된 액터 네트워크

(π_v$(O_v|\theta_v^{\pi})$)가 상대방의 혼합 전략에 대한 최적 순수 전략 최적 응답으로 반환

실험 및 결과

실험 및 결과 – 네트워크 침입 탐지

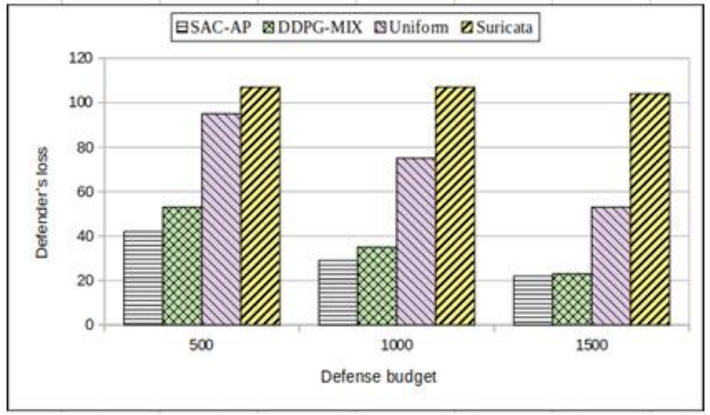


Fig6. 공격 예산 고정, 방어 예산 변화

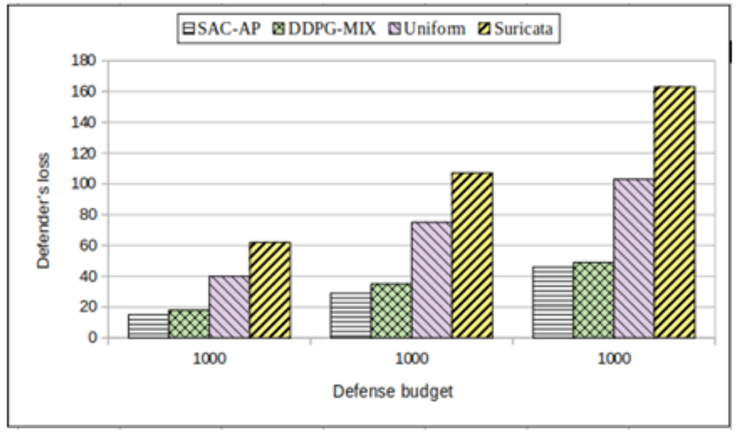


Fig7. 공격 예산 변화, 방어 예산 고정

데이터셋: CICIDS2017 (데이터 약 300만개, 78개 특징 포함)
IDS: Suricata (오픈소스 네트워크 침입 감지 시스템)

Suricata를 통한 이상 탐지 후 SAC-AP를 활용해 우선순위 파악

결과

- DDPG-MIX 및 다른 Baseline 방법 대비 SAC-AP가 월등한 성능
- 방어자의 손실 30% 감소
- 방어자가 공격자의 예산을 모를 때에도 높은 강건성 유지

연구 계획

연구 계획

기존의 이상탐지

- 딥러닝 및 머신러닝을 통한 이상치 탐지를 진행 ex) Auto Encoder, FFT, RL 등
- 이상 탐지에서의 오탐 발생 즉, 이상치로 판별된 데이터 중 정상치 포함
- 이를 보정할 후속 모델의 필요성 증가

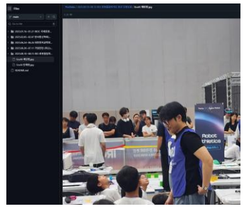
SAC-AP를 활용한 후처리 모델

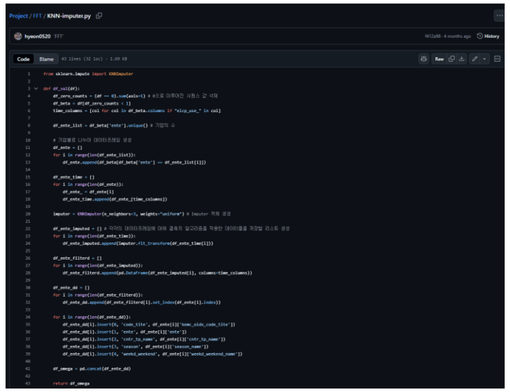
- 경보 우선순위를 기반으로 하는 SAC-AP의 우선순위 선정 알고리즘을 채용
- 이상치에 우선순위를 지정 따라서, 참 이상치를 선정가능
- 방어자(Defender): SAP-AP Agent가 되어 생성된 이상치 중 어떤 것을 조사할지 결정
- 공격자(Adversary): 실제 위협을 유발하는 이상치를 생성하려고 시도
- 상태(State): 현재 시스템의 이상 관련 정보
- 행동(Action): 특정 이상치 또는 이상치 유형을 조사할지 여부, 또는 이상치 탐지 모델의 특정 파라미터를 조정하는 결정
- 보상(Reward): 성공적으로 참 이상치를 탐지하고 조치했을 때 긍정적인 보상, 오탐을 조사하거나 참 이상치를 놓쳤을 때 손실을 받도록 조정

Github

Github

2025.08.10~08.15	IRO 로봇올림피아드 본선	IRO에서 주최한 로봇올림피아드 본선에서 진행요원을 맡아, 참가자 인솔 및 통제를 담당





EcoAI 프로젝트 관련

컴퓨터과학, 프로그래밍, 코딩 관련 프로젝트

Date	Project	Details
2025.05.12~06.24	Fast Furier Transform	고속 푸리에 변환을 활용한 이상치 탐지 관련 논문 및 코드 작성
2025.07.01~	PPO, 강화학습, SAC-AP	강화학습 기반 이상치 탐지 및 후속 연구 탐색

권우현

Activities

기간	활동명	내용
2025.03.21	제44회 데이터 분석 준전문가(ADsP)	자격증 취득
2025.06.24 ~ 06.27	대한전자공학회 하계 종합학술대회	LSTM Autoencoder와 Unsupervised Anomlay Detection 모델의 시간 해상도별 이상치 탐지 성능 분석
2025.06.30 ~ 07.11	(주)젠토 인턴십	허깅페이스의 Text to image/video 모델을 활용한 생성 서비스

이정섭
과거에 활동한 내역 표로 정리
과거 자료 정리 후 더 추가할 예정

Github

Current Projects

- MI-FGSM + FFT 논문 작성 중
- AAAI Student Abstract 작성 중 (강화학습)
- 추계통신학회학술대회 논문 작성 중 (강화학습)

강화학습(RL) 기초 및 Actor-Critic, SAC 정리

1. 강화학습 기본 개념

가치 함수 (Value Function)

에이전트가 특정 정책(π)을 따를 때, 현재 상태(s)에서 앞으로 얻을 수 있는 총 보상의 기댓값을 의미합니다.

$$V_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_t = s\right]$$

- $V_\pi(s)$: 정책 π를 따를 때, 상태 s에서의 가치
- γ^k: 할인 계수(Discount Factor). 미래의 보상을 현재 가치로 환산하며, 0과 1 사이의 값을 가집니다.
- R_{t+k+1}: $t+k+1$ 시점의 보상
- $E_\pi[\cdot \mid s_t = s]$: 현재 상태가 s라는 조건 하에, 정책 π를 따랐을 때의 기댓값

행동 가치 함수 (Action-Value Function / Q-Function)

정책 π를 따를 때, 특정 상태(s)에서 특정 행동(a)을 했을 때 얻을 수 있는 총 보상의 기댓값입니다.

$$Q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_t = s, a_t = a\right]$$

- $Q_\pi(s,a)$: 정책 π를 따를 때, 상태 s에서 행동 a를 했을 때의 가치
- $\mid s_t = s, a_t = a$: 현재 상태가 s이고 행동이 a라는 조건 하에서의 기댓값

정민성
현재 진행 중인 프로젝트 정리
강화학습 알고리즘 개념 정리

감사합니다