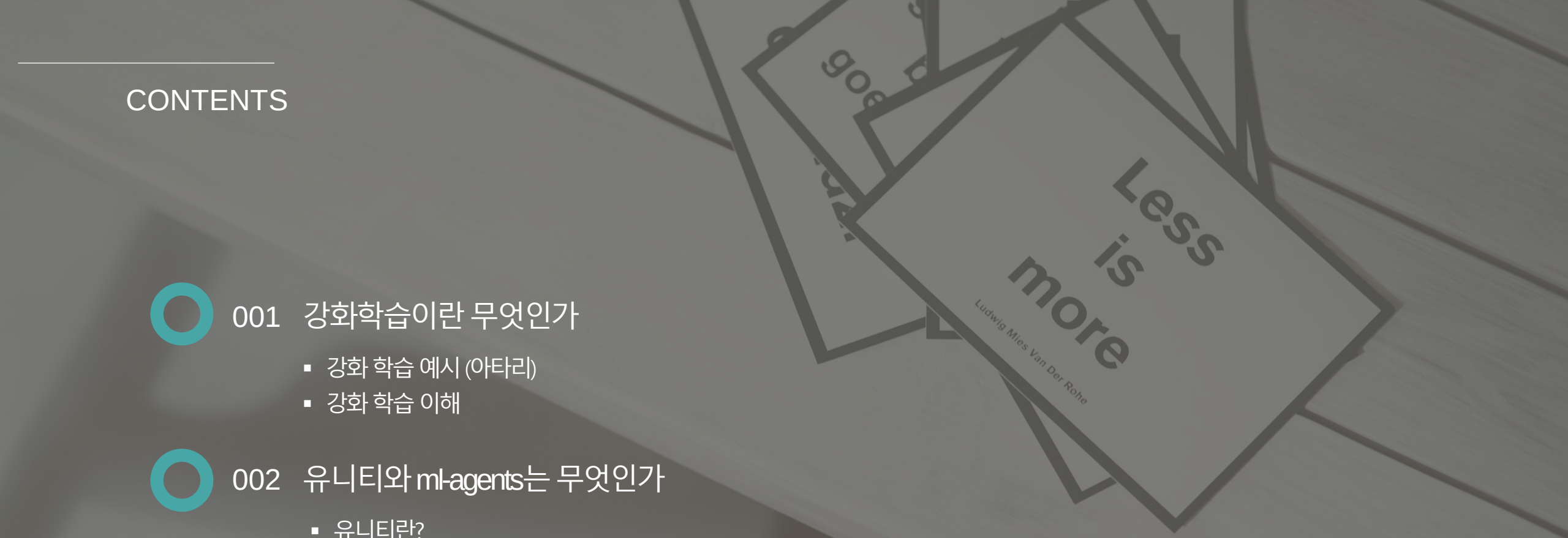
강화학습 이해하기
With Unity

발표자 정현우

2015.07.08

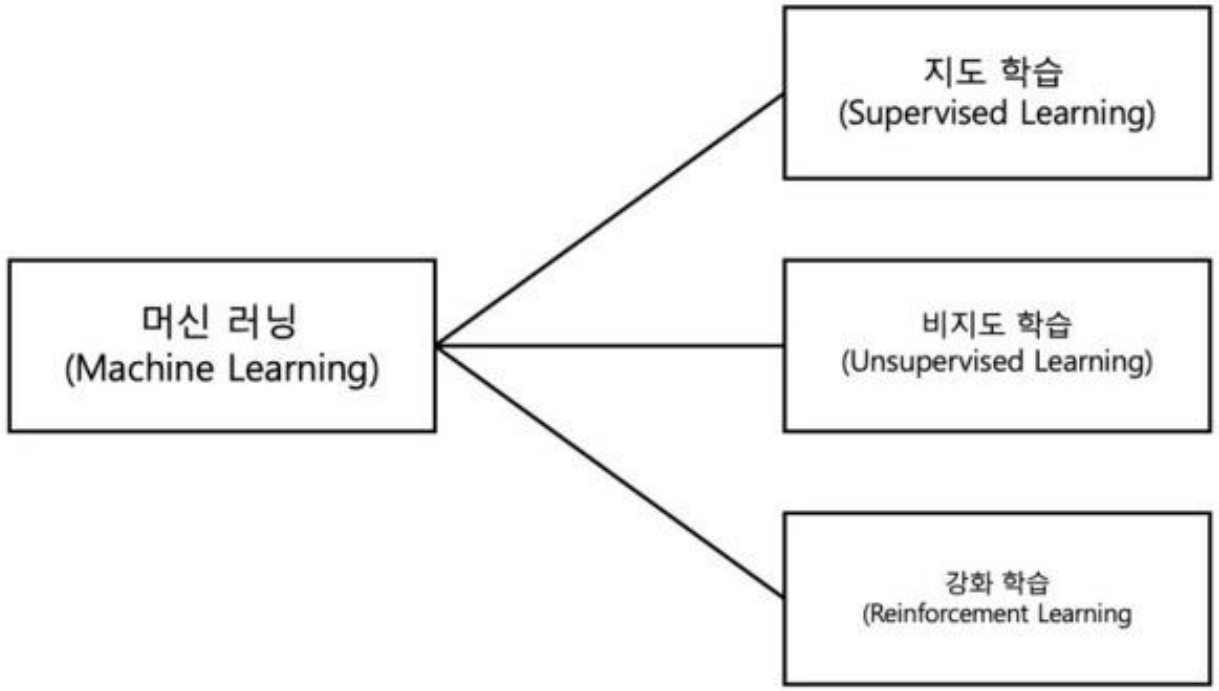
CONTENTS



Part 1.

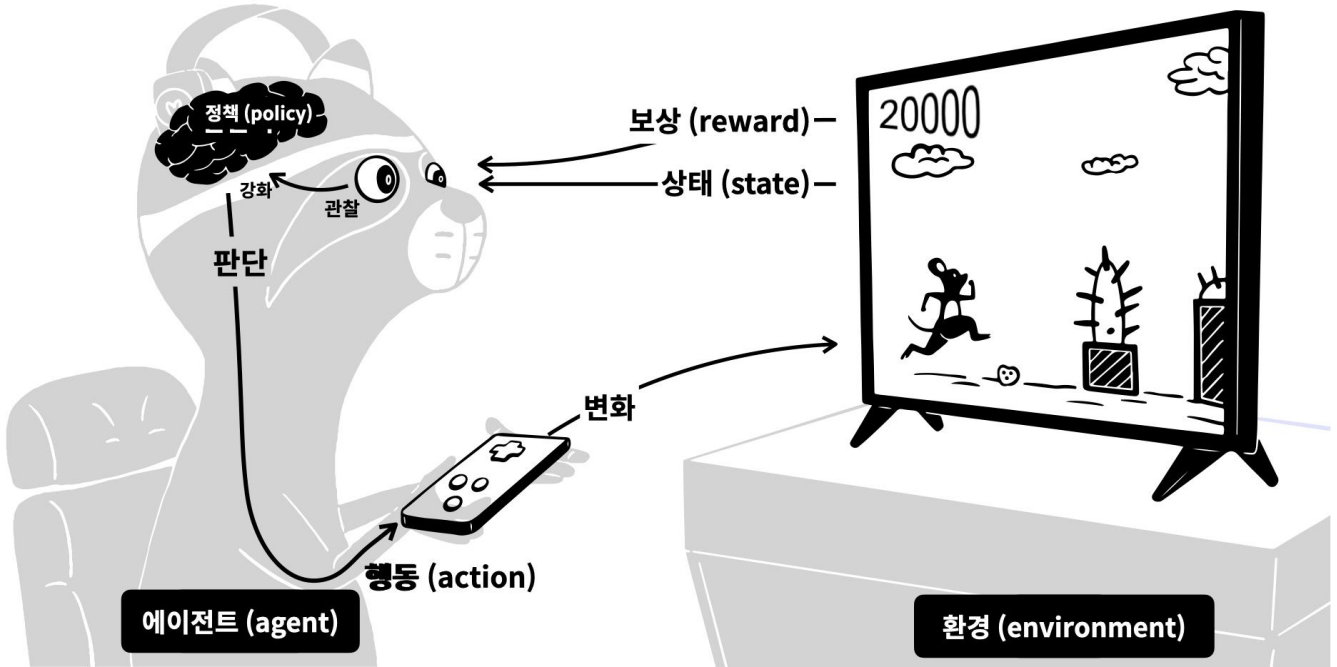
강화학습이란 무엇인가

1.1 강화 학습이란?



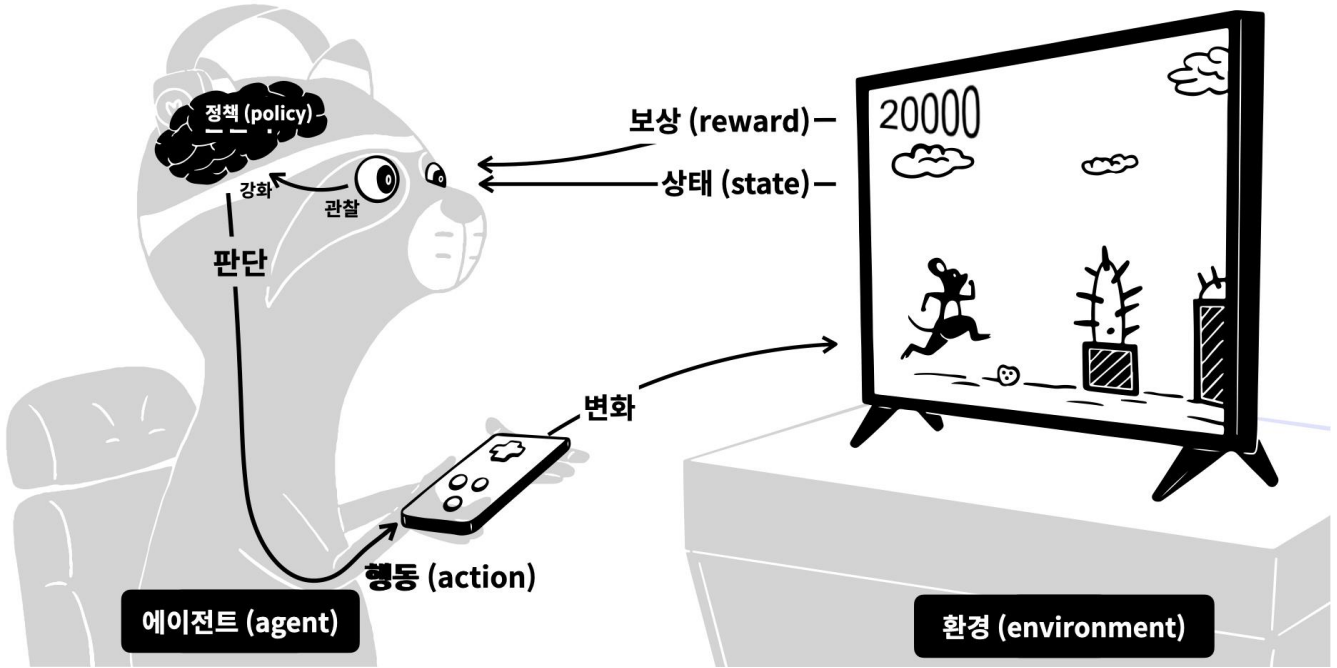
〈머신러닝 분류〉

1.1 강화 학습이란?



- 게임 ➡ 환경(environment)
- 게이머 ➡ 에이전트(agent)
- 게임화면 ➡ 상태(state)
- 게이머의 조작 ➡ 행동(action)
- 상과 벌 ➡ 보상(reward)
- 게이머의 판단력 ➡ 정책(policy)

1.1 강화 학습이란?



참고 : 제 계정이 아니라 인터넷에서 가져온 것 입니다

1.1 강화 학습이란?

1. 게임에는 객관적인 지표들이 있습니다. 내 킬/데스, 내 티어 등을 통해 내가 어느 정도 실력인지 알 수 있습니다.

2. 내 킬/데스를 봅니다. 게임을 잘하고 있다면 팀원들이 따봉을 날릴 것이고, 못하고 있다면 욕설이 날라올 것입니다.

3. 게임을 진행하면서 어떻게 해야 욕을 덜 먹는지 알게 됩니다. 어떻게 해야 팀을 승리로 이끌 수 있는지 생각하게 됩니다.

즉, 판단력이 강화된 것입니다.

4. 판단에 따라서 행동을 합니다.

5. 그 행동은 게임에 변화를 주게 됩니다.

참고 : 제 계정이 아니라 인터넷에서 가져온 것 입니다

1.1 강화 학습이란?

비유하자면 지도학습이 배움을 통해서
실력을 키우는 것이라면,
강화학습은 일단 해보면서 경험을 통해서
실력을 키워가는 것입니다.

그 행동의 결과가 자신에게 유리한 것이었다면
상을 받고, 불리한 것이었다면 벌을 받는 것입니다.

이 과정을 매우 많이 반복하면
더 많은 보상을 받을 수 있는 더 좋은 답을
찾아낼 수 있다는 것이 강화학습의 기본 아이디어입니다.

[생활 코딩 - 강화학습]

1.1 강화 학습이란?

예시: 벽돌깨기 고수

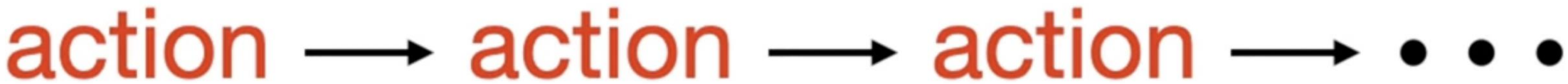
처음엔 멍청하지만

시간을 들여 학습을 진행할 수록

놀라운 성능을 보여준다.

어찌 보면 인간보다 더 뛰어난 성능을 보인다.

1.2 강화 원리



Goal = maximize Expected Return

1.2 Q-learning

기본 원리 예시

맛집을 찾고 싶은데
나는 길치인데다가
핸드폰도 지금 없네...
어떻게 해야 맛집을 잘
찾아갈까?

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,0) ,(0,0)** **(좌,우), (상,하)**			**맛집** **(R=1)**

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,0) ,(0,0)** **(좌,우), (상,하)**			**맛집** **(R=1)**
			(0,0),(1,0) 아무렇게나 갔더니 맛집을 찾았다!! (정보 저장)

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,0) ,(0,0)** **(좌,우), (상,하)**			**맛집** **(R=1)**
		(0,1), (0,0) 여기서 오른쪽으로 우연히 갔다. (이 정보(가장 큰 값)도 저장)	(0,0),(1,0) 맞다 여기서 위로 가면 맛집이었지?

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,0) ,(0,0)** **(좌,우), (상,하)**			**맛집** **(R=1)**
	(0,1),(0,0) 어쩌다 왔더니 오른쪽에 기억해 놓은 길이 있어서 업데이트	(0,1), (0,0)	(0,0),(1,0)

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,0) ,(0,0)** **(좌,우), (상,하)**			**맛집** **(R=1)**
(0,1),(0,0)	(0,1),(0,0)	(0,1), (0,0)	(0,0),(1,0)

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,0) ,(0,1)** **(좌,우), (상,하)**			**맛집** **(R=1)**
(0,1),(0,0)	(0,1),(0,0)	(0,1), (0,0)	(0,0),(1,0)

근데 직선으로 가는데 더 빠르지 않을까?

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0,1) ,(0,1)** **(좌,우), (상,하)**	**(0,1),(0,0)**	**(0,1),(0,0)**	**맛집** **(R=1)**
(0,1),(0,0)	(0,1),(0,0)	(0,1), (0,0)	(0,0),(1,0)

어? 둘다 값이 1로 같으면 어딜 가야 하는거야?

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0, γ^2) ,(0, γ^4)** **(좌,우), (상,하)**	**(0,γ),(0,0)**	**(0,1),(0,0)**	**맛집** **(R=1)**
(0, γ^3),(0,0)	(0, γ^2),(0,0)	(0, γ), (0,0)	(0,0),(1,0)

감마를 도입해보자!! (γ 는 0~1사이의 값)

1.2 Q-learning

강화 학습의 기본 원리

시작 지점 **(0, γ^2) ,(0, γ^4)** **(좌,우), (상,하)**	**(0,γ),(0,0)**	**(0,1),(0,0)**	**맛집** **(R=1)**
(0, γ^3),(0,0)	(0, γ^2),(0,0)	(0, γ), (0,0)	(0,0),(1,0)
			진짜 진짜 맛집 (R=10)

더 맛집이 있으면?
너무 길 따라 가면 진짜 맛집을 못 찾을 수 있으니
입실론 값을 점차 줄이기(탐험)

1.2 Q-learning

식으로 표현해보자!

$$Q(s_t, a_t) \leftarrow (1-\alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} \right)$$

각 시간 t에서 어떠한 상태 S(t)에서 행동 a(t)를 취하고 새로운 상태 S(t+1)로 전이한다.

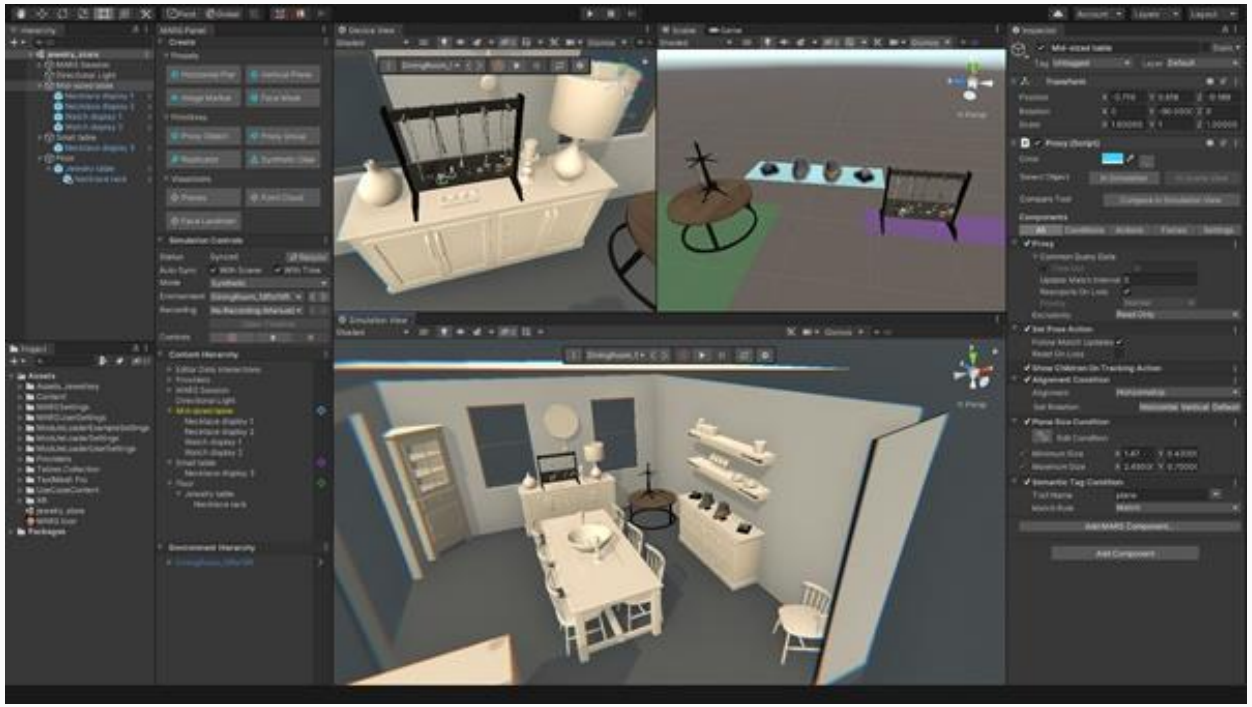
이 때 보상 r(t)가 얻어지며 Q함수가 갱신된다.

Learning rate는 새로운 정보를 얼마나 받아들이냐 이다.

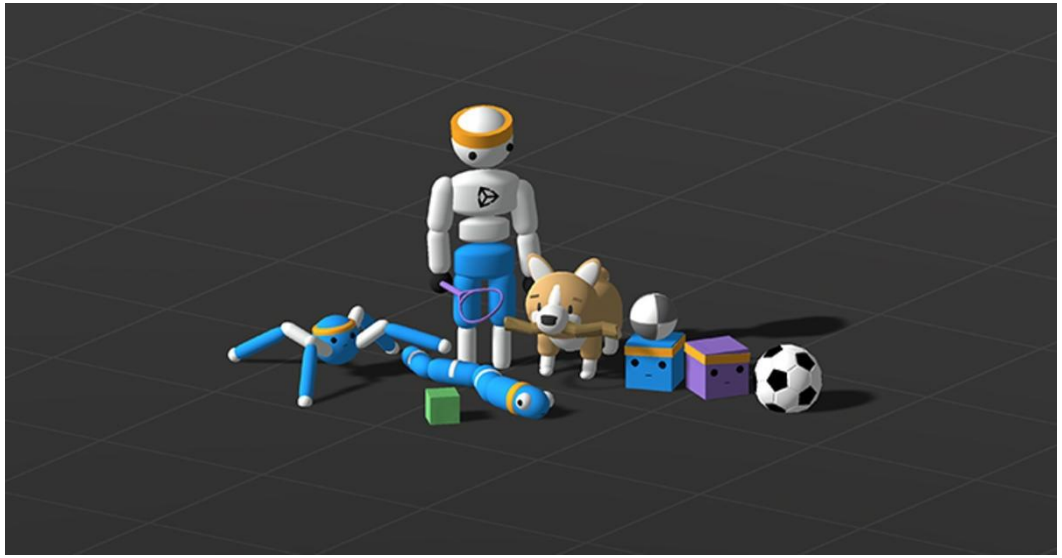
Part 2.

유니티와 ml-agents는 무엇인가

2.1 유니티와 ml-agents란 무엇인가?



2.1 유니티와 ml-agents란 무엇인가?



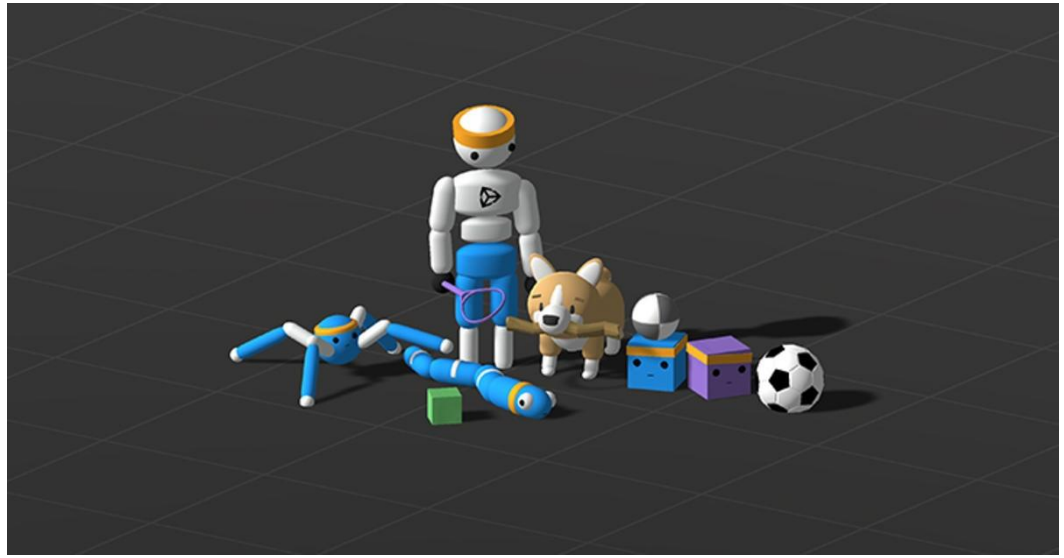
모델의 학습을 위한 현실적이고 복합적인 AI 환경 구현하기

인공지능(AI) 연구가 발전하려면 현재의 AI 모델 학습용 벤치마크를 사용하여 기존 환경에 존재하는 어려운 문제를 해결해야만 합니다. 하지만 이러한 문제가 '해결'되고 나면 새로운 환경의 필요성이 대두되곤 합니다. 하지만 그러한 환경을 조성하려면 많은 시간과 전문적이고 특별한 지식이 필요한 경우가 많습니다.

Unity와 ML-Agents 툴킷을 사용하면 풍부한 물리적, 시각적, 인지적 요소를 갖춘 AI 환경을 조성할 수 있습니다. 새로운 알고리즘과 메서드의 연구는 물론이고 벤치마킹에도 이러한 환경을 사용할 수 있습니다.

[유니티 공식 사이트]

2.1 유니티와 ml-agents란 무엇인가?



MI-agent가 가지는 장점은 무엇일까?

1. **오픈 소스**
Unity ML-Agents 툴킷은 Apache 2.0 라이선스가 적용된 오픈 소스입니다. 이 툴킷을 사용하면 필요에 따라 ML-Agents를 수정하고 구현할 수 있습니다.

2. **AI/ML 전문 지식 불필요**
툴킷에는 바로 사용할 수 있는 첨단 알고리즘과 탄탄한 문서 및 예제 프로젝트 등 시작에 필요한 모든 항목이 포함됩니다. 또한 유용한 게임 개발 커뮤니티의 지원도 받을 수 있습니다.

3. **최소한의 코딩으로 간편한 설정**
게임을 AI 학습 환경으로 쉽고 빠르게 설정할 수 있습니다. 대량 코딩 작업 없이 지능형 캐릭터를 제작할 수 있습니다.

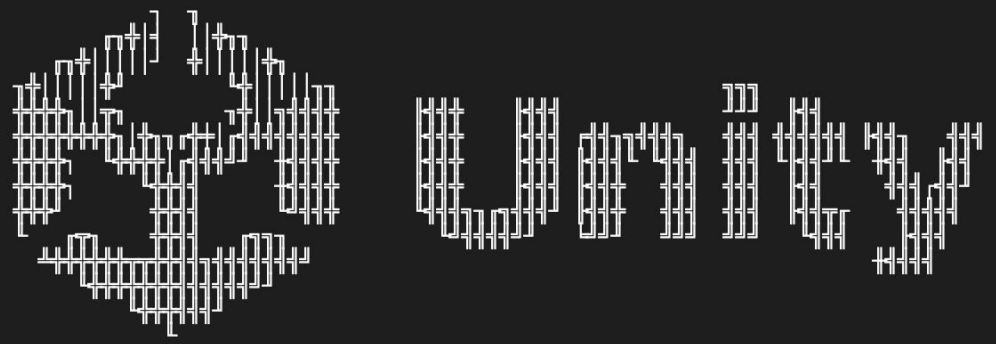
[유니티 공식 사이트]

Part 3.

MI agent 실습

3.1 실습 오류시 자료

```
[(base) jhw@jeonghyeon-uui-MacBookPro ml-agents % conda activate py36
[(py36) jhw@jeonghyeon-uui-MacBookPro ml-agents % mlagents-learn config/ppo/3DBall.yaml --run-id=first3DBallRun
```



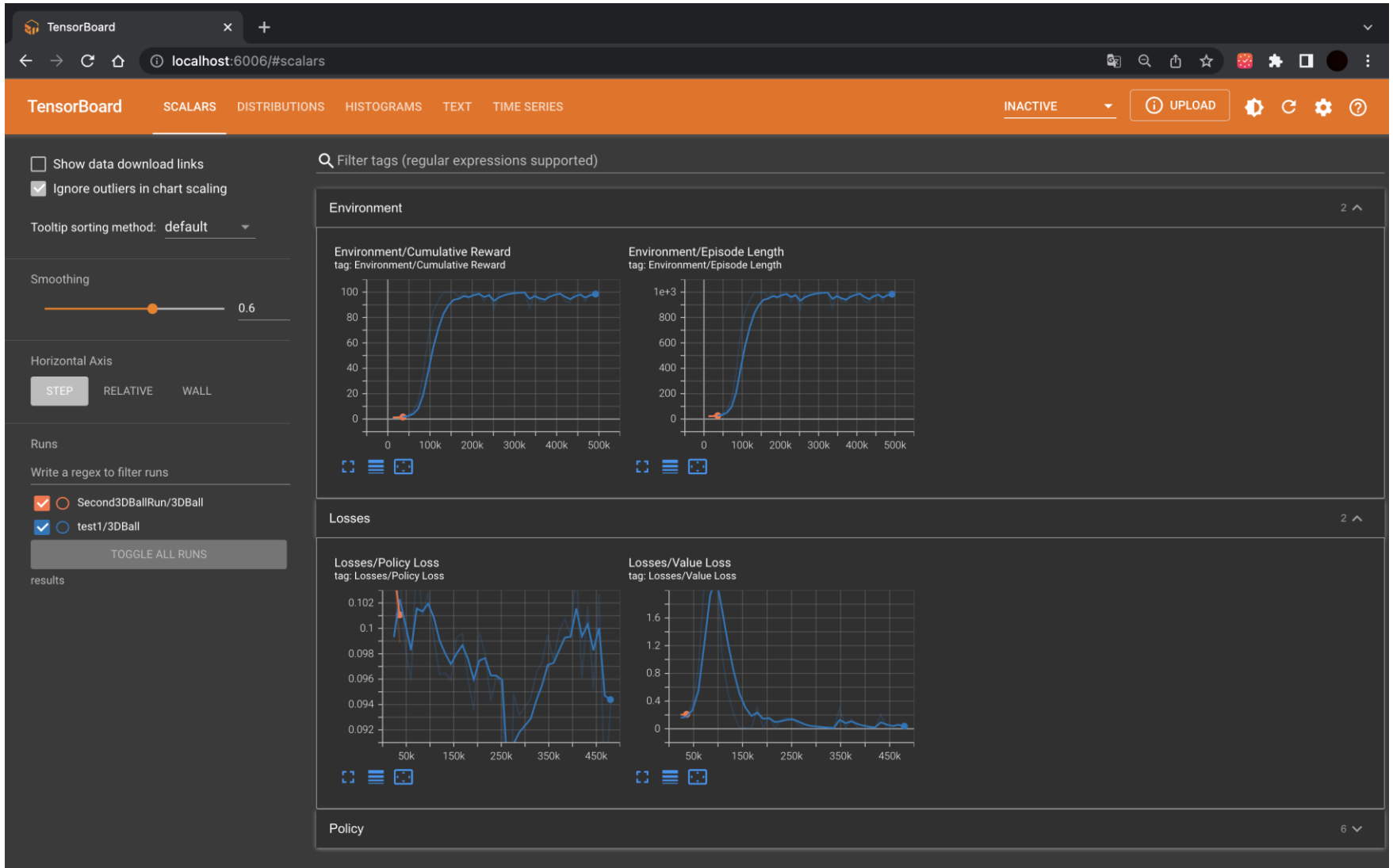
```
 Version information:
  ml-agents: 0.28.0,
  ml-agents-envs: 0.28.0,
  Communicator API: 1.5.0,
  PyTorch: 1.7.1
Traceback (most recent call last):
  File "/Users/jhw/opt/anaconda3/envs/py36/bin/mlagents-learn", line 33, in <module>
    sys.exit(load_entry_point('mlagents', 'console_scripts', 'mlagents-learn')())
  File "/Users/jhw/ml-agents/ml-agents/mlagents/trainers/learn.py", line 260, in main
    run_cli(parse_command_line())
  File "/Users/jhw/ml-agents/ml-agents/mlagents/trainers/learn.py", line 256, in run_cli
    run_training(run_seed, options, num_areas)
  File "/Users/jhw/ml-agents/ml-agents/mlagents/trainers/learn.py", line 75, in run_training
    checkpoint_settings.maybe_init_path,
  File "/Users/jhw/ml-agents/ml-agents/mlagents/trainers/directory_utils.py", line 26, in validate_existing_directories
    "Previous data from this run ID was found. "
mlagents.trainers.exception.UnityTrainerException: Previous data from this run ID was found. Either specify a new run ID, use --resume to resume this run, or use the --force parameter to overwrite existin
g data.
```

3.1 실습 오류시 자료

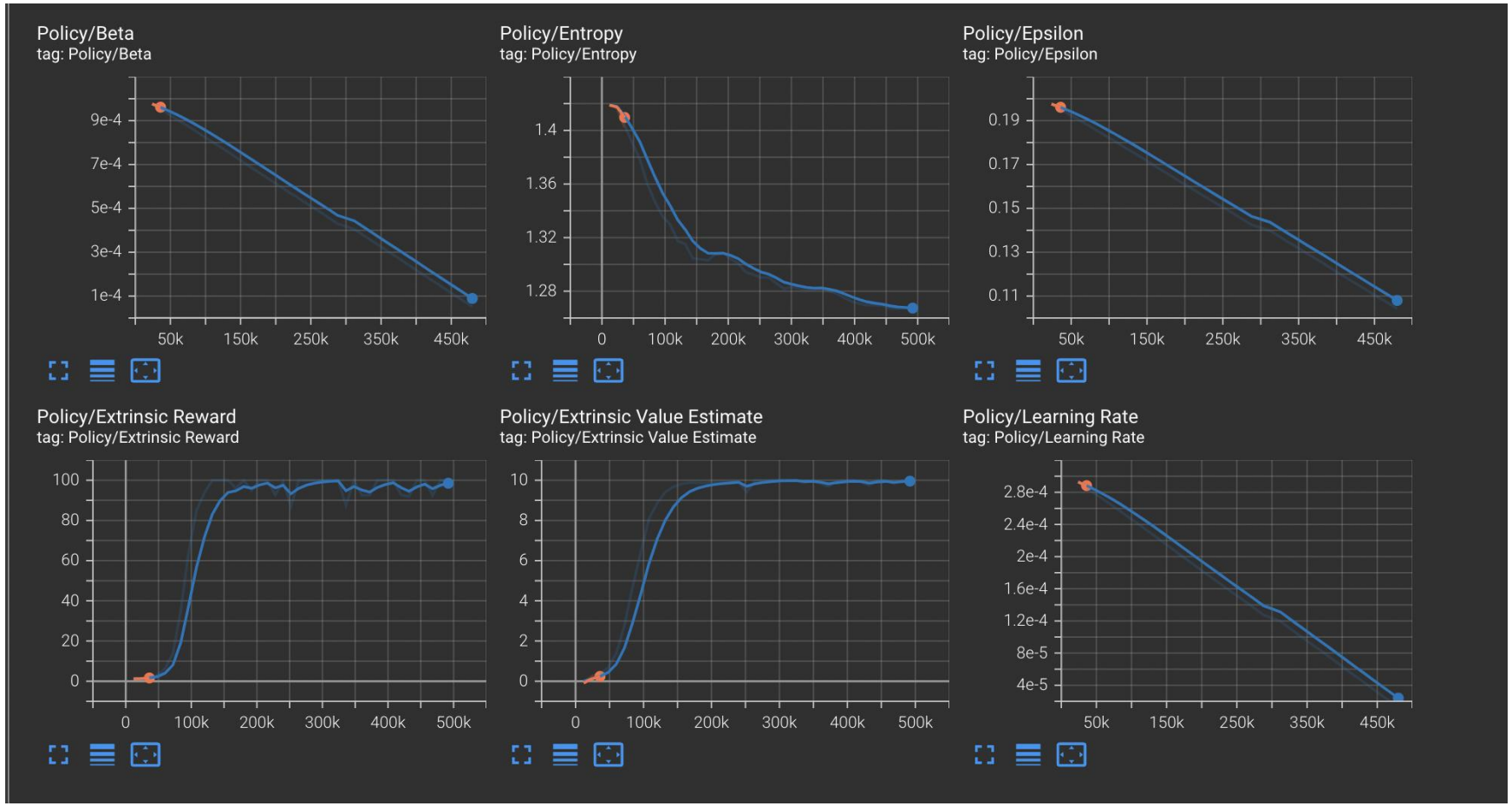
3.1 실습 오류시 자료

```
[INFO] Listening on port 5004. Start training by pressing the Play button in the Unity Editor.
[INFO] Connected to Unity environment with package version 2.2.1-exp.1 and communication version 1.5.0
[INFO] Connected new brain: 3DBall?team=0
[INFO] Hyperparameters for behavior name 3DBall:
        trainer_type:   ppo
        hyperparameters:
          batch_size:   64
          buffer_size:  12000
          learning_rate:        0.0003
          beta: 0.001
          epsilon:      0.2
          lambd:        0.99
          num_epoch:    3
          learning_rate_schedule:       linear
          beta_schedule:        linear
          epsilon_schedule:     linear
        network_settings:
          normalize:    True
          hidden_units: 128
          num_layers:   2
          vis_encode_type:      simple
          memory:       None
          goal_conditioning_type:       hyper
          deterministic:        False
        reward_signals:
          extrinsic:
            gamma:      0.99
            strength:   1.0
            network_settings:
              normalize:        False
              hidden_units:     128
              num_layers:       2
              vis_encode_type:  simple
              memory:   None
              goal_conditioning_type:   hyper
              deterministic:    False
        init_path:      None
        keep_checkpoints:       5
        checkpoint_interval:    500000
        max_steps:      500000
        time_horizon:   1000
        summary_freq:   12000
        threaded:       False
        self_play:      None
        behavioral_cloning:     None
```

3.1 실습 오류시 자료



3.1 실습 오류시 자료



Part 4.

Reference

4.1 참고자료

> https://unity.com/kr/products/machine-learning-agents
> https://www.youtube.com/watch?v=3Ch14GDY5Y8&ab_channel=%ED%98%81%ED%8E%9C%ED%95%98%EC%9E%84

감사합니다

발표자 정현우