Reinforcement Learning for Generalization

2021-03-12

Data **M**ining and **Q**uality **A**nalytics Lab

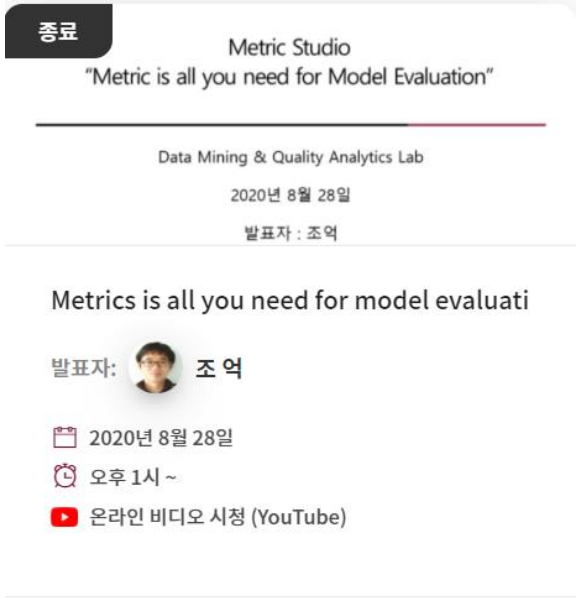
조억

발표자 소개

◈ 조억

- E-mail : ukjo@korea.ac.kr
- Data Mining & Quality Analytics Lab(김성범 교수님)
- 석박사 통합과정 4학기 재학중(2019.8 ~)

관심분야
- Reinforcement Learning
- Time Series Data Analysis



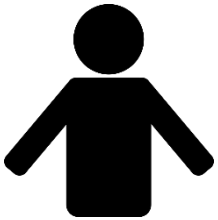
DMQA

목차

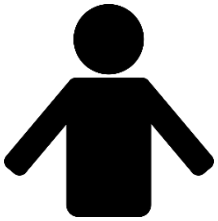
DMQA

목적

- 강화학습 기본 원리 이해
- 기존 강화학습 문제점
- 메타 강화학습을 통한 한계점 극복 방안
- 각 극복 방안의 대표적인 알고리즘에 대한 개념적 이해



강화학습 기본



강화학습 문제점

메타 강화학습
다양한 접근법

※ Model Free Reinforcement Learning과 Finite MDP 가정하고 진행됩니다

Reinforcement Learning

Reinforcement Learning

환경(Environment)과 상호작용을 하는 **에이전트(Agent)는 상태(State)**를 보고 **액션(Action)**하며, 그에 대한 응답으로 **보상(Reward)**을 얻는 연속적 의사 결정

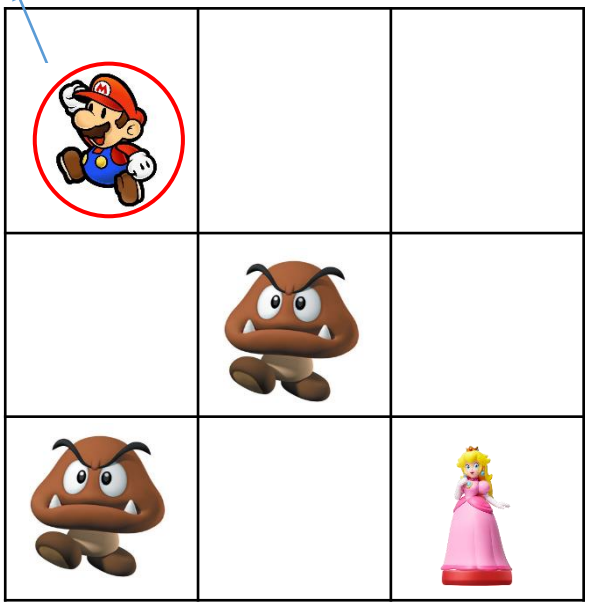
DMQA

Reinforcement Learning

Environment

(Grid World → Mario World)

에이전트(Agent)



상태(State)

: {
  {에이전트, 0, 0},
  {0, 굼바, 0},
  {굼바, 0, 공주}
}

액션(Action)

: {위, 아래, 좌, 우}

리워드(Reward)

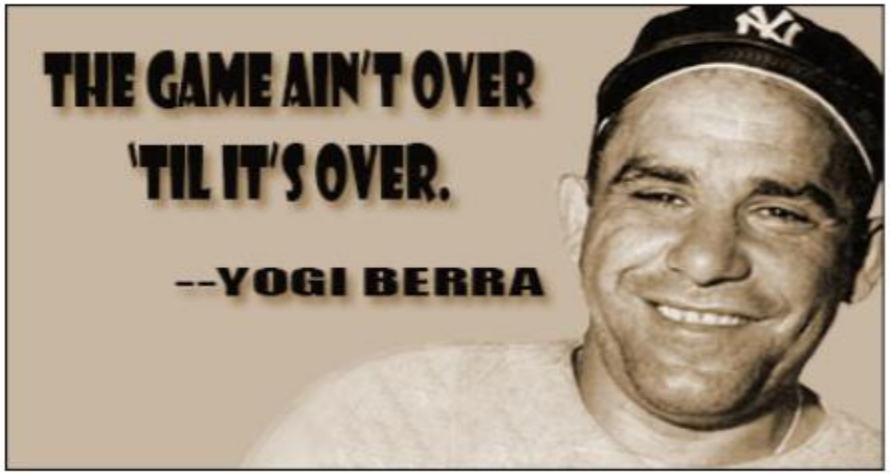
 : +1 ,  : -1

Reinforcement Learning

에이전트 목표는 처음 시작하는 시점부터 종료시점까지 일어나는 모든 에피소드에서 받을 보상값 최대로 끌어올리는 것

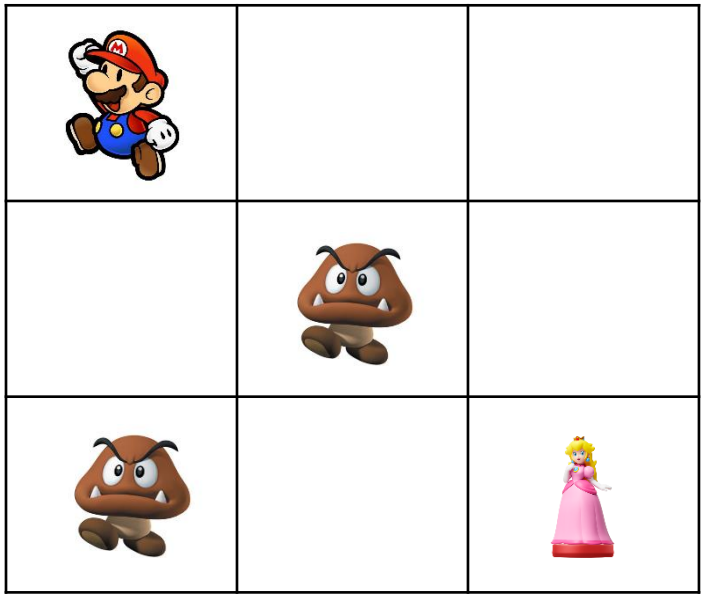
"끝날 때까지 끝난 게 아니다.(It ain't over till it's over)"

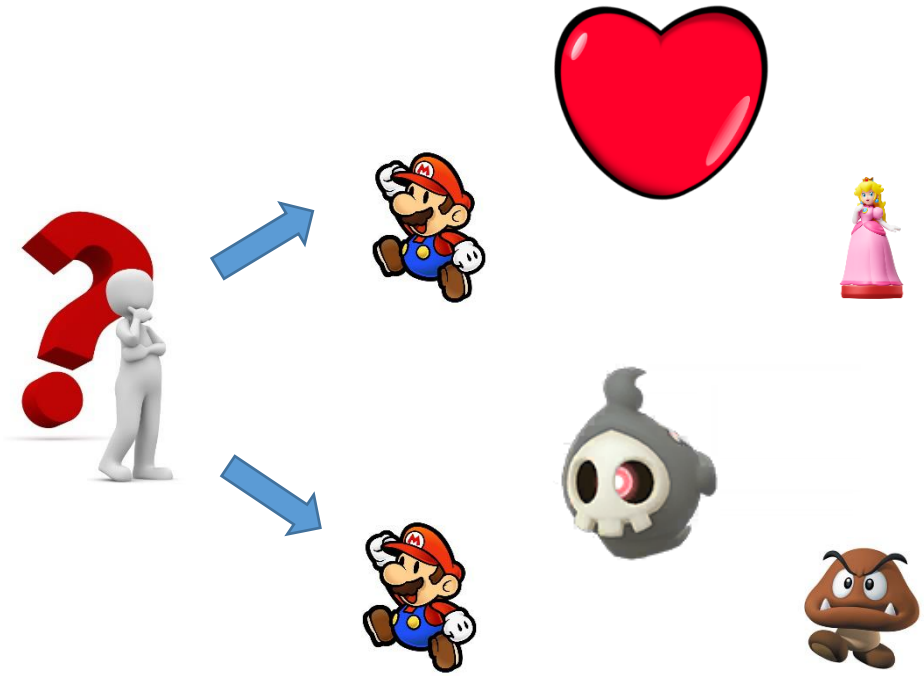
스포츠 해설을 듣다 보면 자주 등장하는 말. 미국의 유명한 야구선수 요기 베라(Yogi Berra)가 남긴 명언이다. '결과는 아무도 모르니 끝까지 최선을 다하자'는 뜻으로, 마지막까지 열심히 노력하자는 교훈을 준다.



Reinforcement Learning

에이전트가 시작시점에서 게임 끝날 때 최종 시점의 점수를 최대화하는 방법을 어떻게 알 수 있을까?





Markov Decision Process(MDP)

연속적 의사 결정 문제를 해결하기 위해 강화학습에서 사용하는 프로세스

$$\mathcal{MDP} = < S, A, P, R, \gamma >$$

S : 상태(State)
A : 액션(Action)
P : 상태변환확률(State transition probability)
R : 보상(Reward)
γ : 할인율(Discount factor)

가치함수(Value Function)

$$Q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

정책(Policy)

$$\pi(a|s) = P(A_t = a | S_t = s)$$

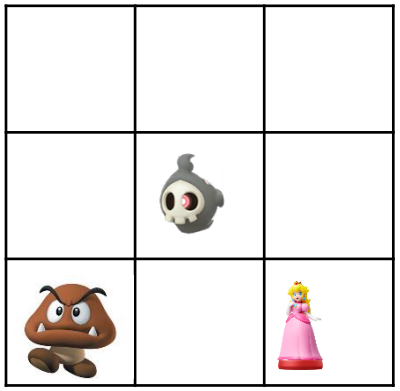
누적 보상(Return)

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{T} \gamma^k R_{t+k+1}$$

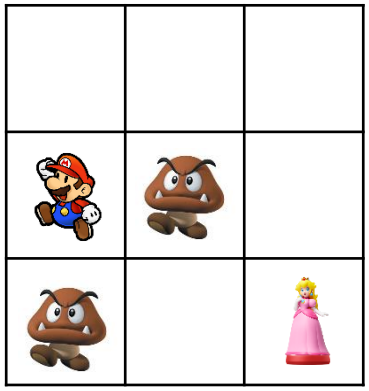
Markov Decision Process(MDP)

$P^{a}_{ss'}$: 상태변환확률(State transition probability)

S_3 : 세번째 상태



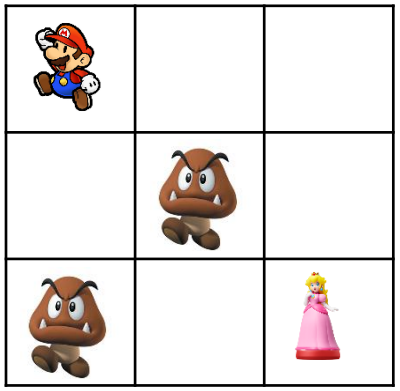
S_2 : 두번째 상태



20%

80%

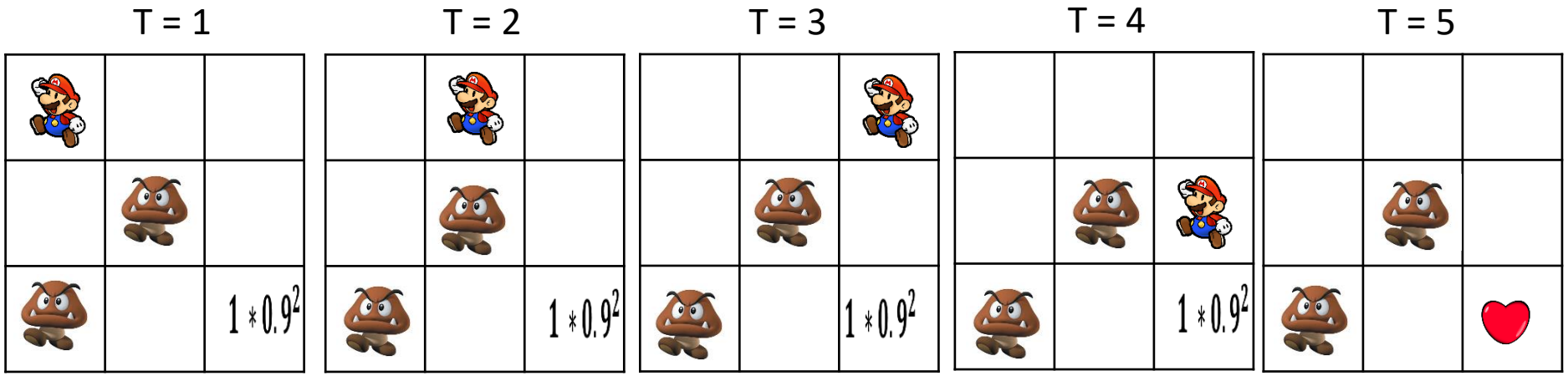
S_3 : 세번째 상태



a_2 = 위

Markov Decision Process(MDP)

γ : 할인율(Discount factor)



$1 * 0.9^4$ $1 * 0.9^3$ $1 * 0.9^2$ $1 * 0.9^1$ 1

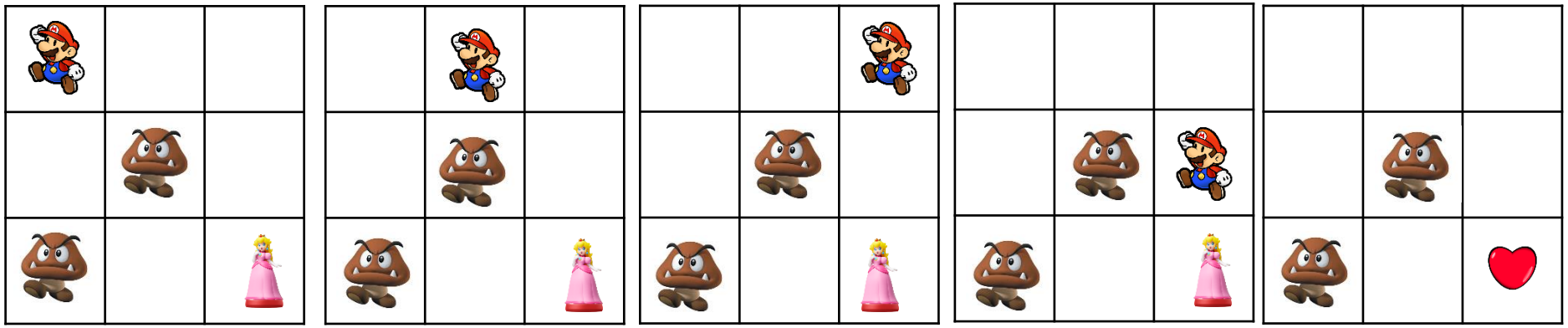
Reward를 받은 최종 시점에서부터 멀어질수록 리워드 할인

DMQA

Markov Decision Process(MDP)

G_t : 누적 보상, 강화학습 에이전트가 최대화 해야 하는 값 (Return)

T = 1 T = 2 T = 3 T = 4 T = 5



$G_t = 1 * 0.9^4$ $1 * 0.9^3$ $1 * 0.9^2$ $1 * 0.9^1$ 1

$$G_t = R_{t+1} + \gamma R_{(t+2)} + \ \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

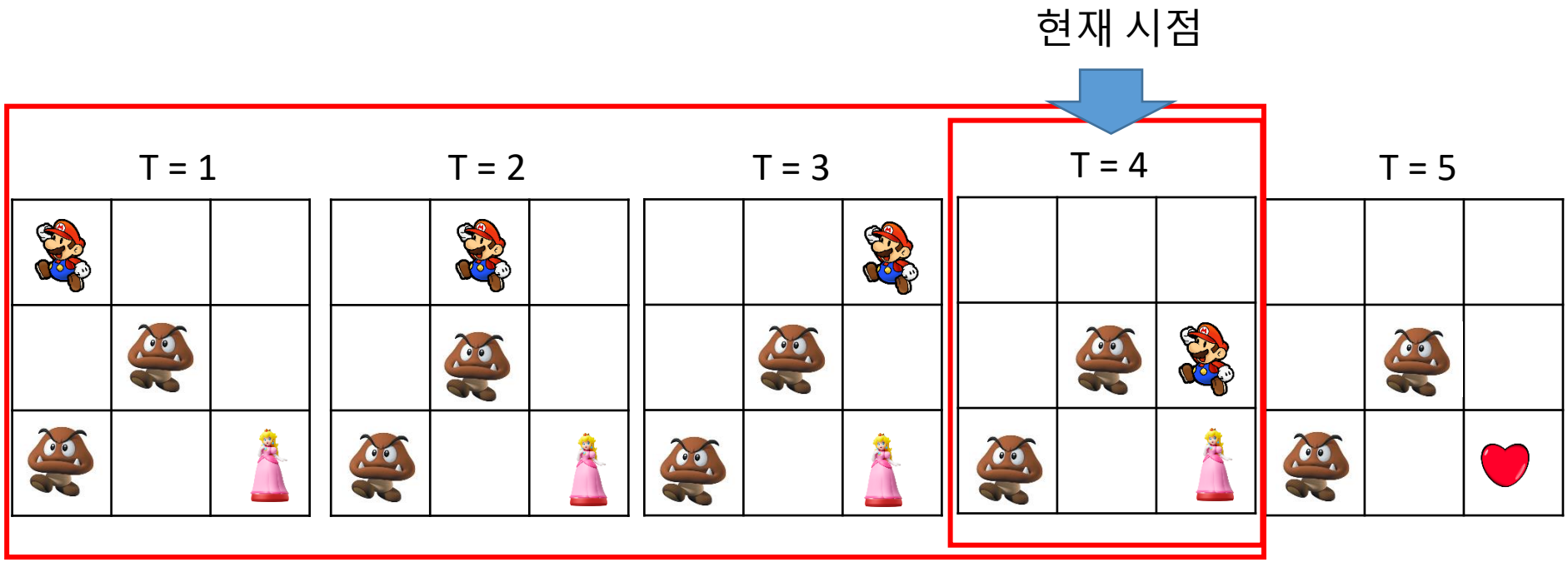
Markov Decision Process(MDP)

Markov Property (=memoryless property)

= 미래는 오직 과거의 정보가 아닌 현재의 정보에만 의존한다.

= 이전의 상태정보를 감안하지 않으므로 학습 효율적

$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, S_2, \dots, S_t]$$



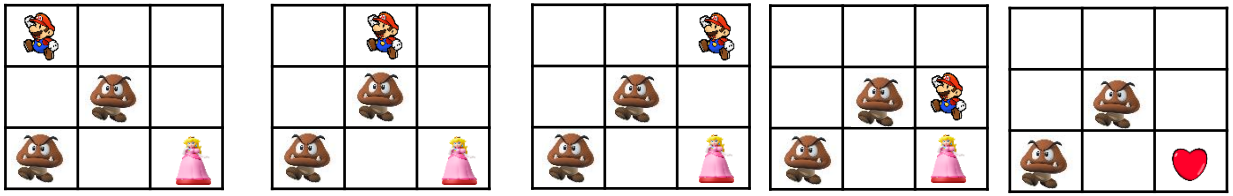
DMQA

Markov Decision Process(MDP)

가치 함수(Value Function) : 상태(State)마다 누적 보상 기대값(평균 가치) 함수

$$v(s) = \mathbb{E}[G_t | S_t = s]$$

#1



#2

#3

-0.9639	0.729	0.81
-1.8	-1	0.9
-1	0	1

v

Markov Decision Process(MDP)

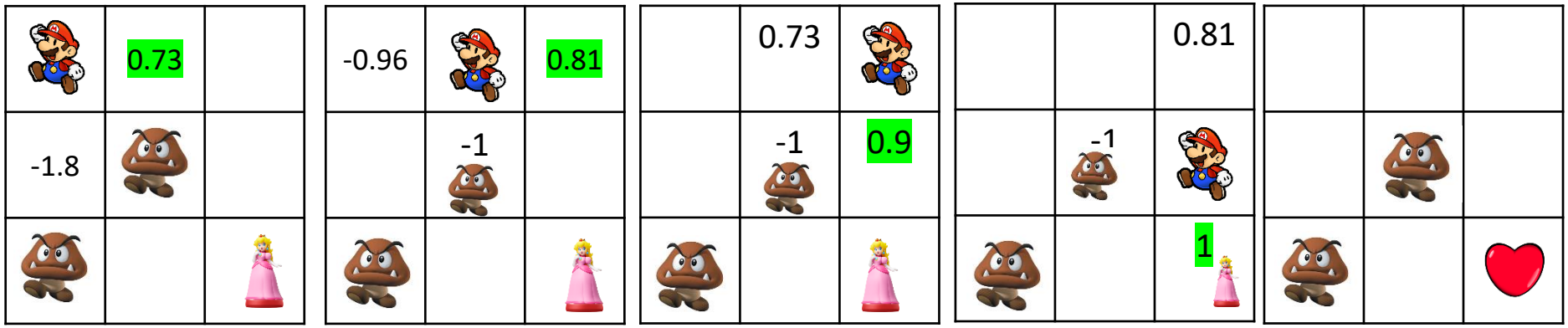
정책(Policy) : 각 상태에서 에이전트가 액션을 결정해주는 함수

$$\pi(a|s) = P(A_t = a \mid S_t = s)$$

-0.96	0.73	0.81
-1.8	-1	0.9
-1	0	1

$\pi(a|s)$

-0.96	0.73	0.81
-1.8	-1	0.9
-1	0	1



DMQA

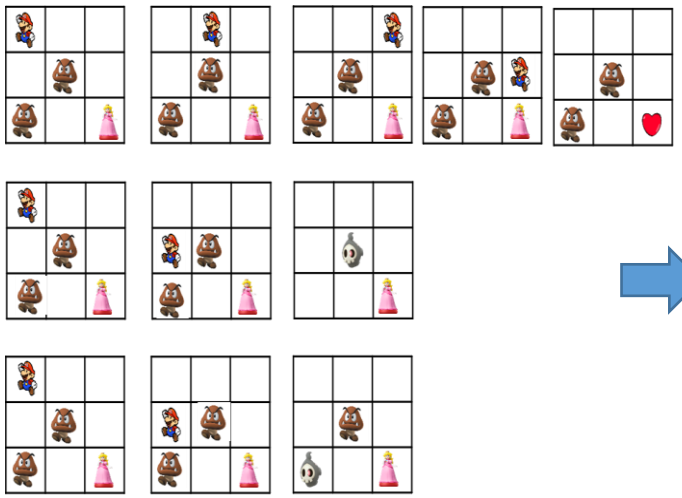
Exploration & Exploitation

에이전트가 목적을 달성하기 위해 가치 함수는 다양한 State 와 Action 조합이 필요

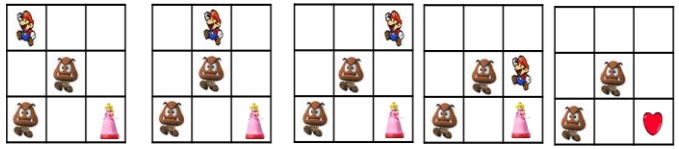
초반에는 Exploration, 학습 잘된 이후엔 Exploitation!!

Exploration

Exploitation

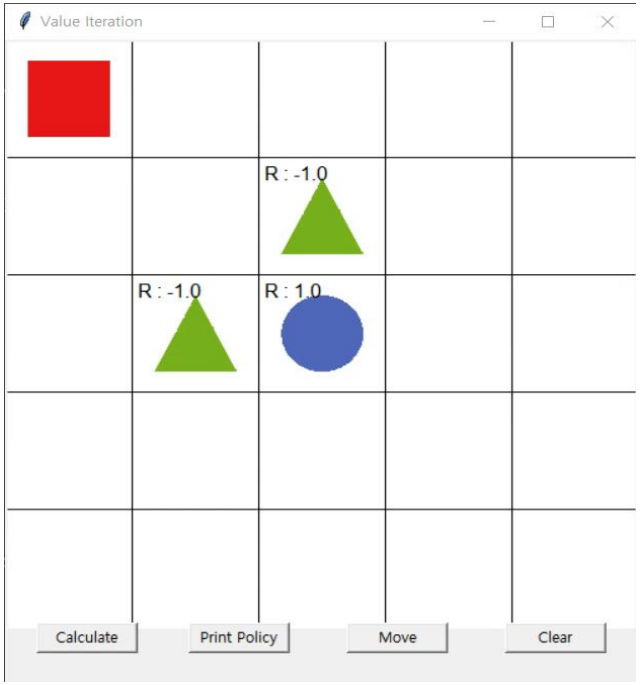


-0.96	0.73	0.81
-1.8	-1	0.9
-1	0	1



Reinforcement Learning

마리오가 공주를 만나기 위해 에이전트가 학습되는 과정



https://github.com/rlcode/reinforcement-learning-kr/

DMQA

Reinforcement Learning

더 자세한 내용은 아래 제 이전 세미나 페이지 1 ~54를 참고바랍니다.

Cooperative Multi-Agent Reinforcement Learning Framework for Scalping Trading

2020년 1월 10일 오후 5:15 / 조회수: 1254

REFERENCES

Cooperative Multi-Agent Reinforcement Learning Framework for Scalping Trading-조억_v1.0.pdf

INFORMATION

2020년 1월 17일 오후 1시 ~ 고려대학교 신공학관 218호

발표자: 조 억

TOPIC

Cooperative Multi-Agent Reinforcement Learning Framework for Scalping Trading

OVERVIEW

요약: 강화학습이 많은 게임에서 그 성능을 입증하며 이제는 현실문제를 푸는 중요한 수단이자 프레임워크로 급부상하고 있다. 이 연구에서는 한국 주식시장에서 수집한 틱 단위의 데이터를 가지고 스켈핑이라고 하는 초단타 매매를 하는 거래 상황을 강화학습 환경으로 정의하였고 이 환경속에서 Deep Q Network으로 구성된 4가지의 에이전트로 매수신호, 매수, 매도 신호, 매도를 각각 수행하게 역할분담하여 에이전트들간 협업을 통하여 수익을 최대화하는 미션을 수행하고자 하였다. 이 연구에서는 초단타 매매를 위한 현실에 가깝게 만든 환경 및 에이전트를 통하여 현실에 적용이 가능하도록 에이전트 상태와 액션에 정의를 한 부분에 대해서 다루며, 마지막으로 실험 결과로 강화학습의 주식시장에서의 적용 가능성을 엿보고자 한다.

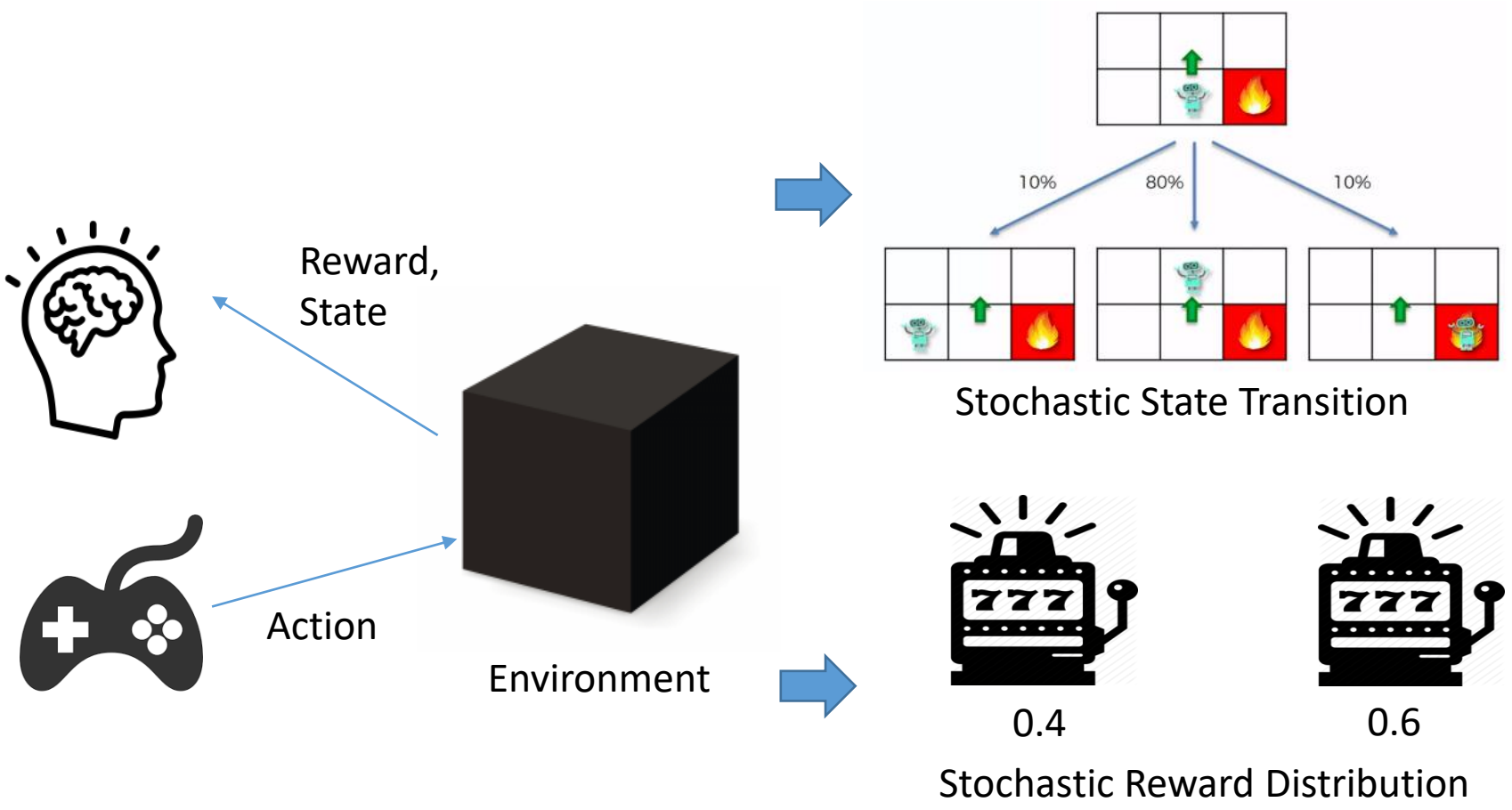
참고논문: Jo, U., Jo, T., Kim, W., Yoon, I., Lee, D., & Lee, S. (2019). Cooperative Multi-Agent Reinforcement Learning Framework for Scalping Trading. arXiv preprint arXiv:1904.00441.

http://dmqm.korea.ac.kr/activity/seminar/277

Traditional RL Problems

Environment Dynamics

에이전트가 환경으로부터 학습하는 목적은 다음과 같다.

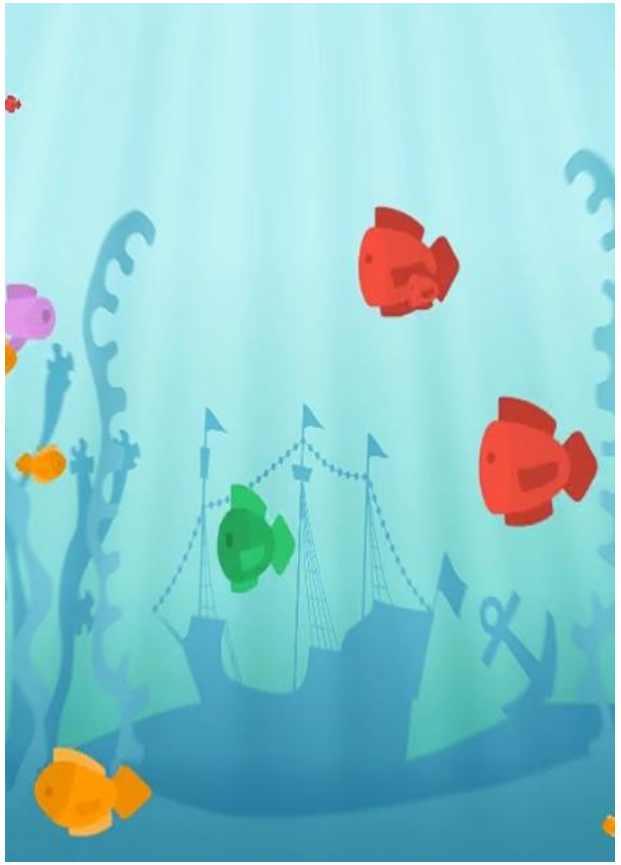


Traditional RL Problems

이미지(RGB)정보를 상태로 받는 에이전트가 테스트 시점에 배경이 바뀐다면?

Training

Test



Traditional RL Problems

슬롯 머신의 성공 확률이 테스트시점에서 바뀐다면?

Training

0.3

0.7

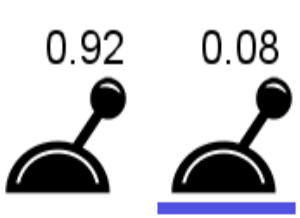
Test

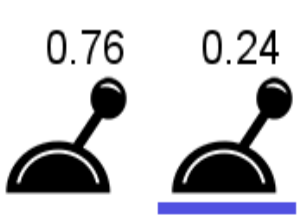
0.6

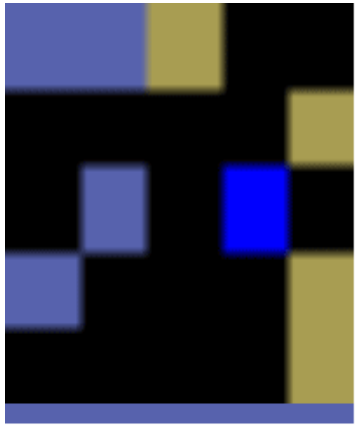
0.4

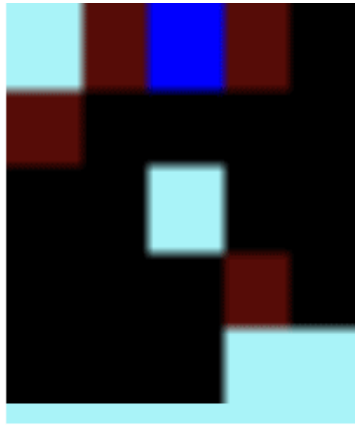
Traditional RL Problems

학습되는 환경 밖에 모르는 에이전트는 테스트 시점에서는 성능을 내지 못한다.









Step: 0

Reward: 0.0

Step: 0

Reward: 0.0

Trial: 1

Trial: 1

Meta Reinforcement Learning

Meta Reinforcement Learning

메타 러닝과 강화학습을 접목하여 강화학습의 일반적인 성능을 올리는 학습방법

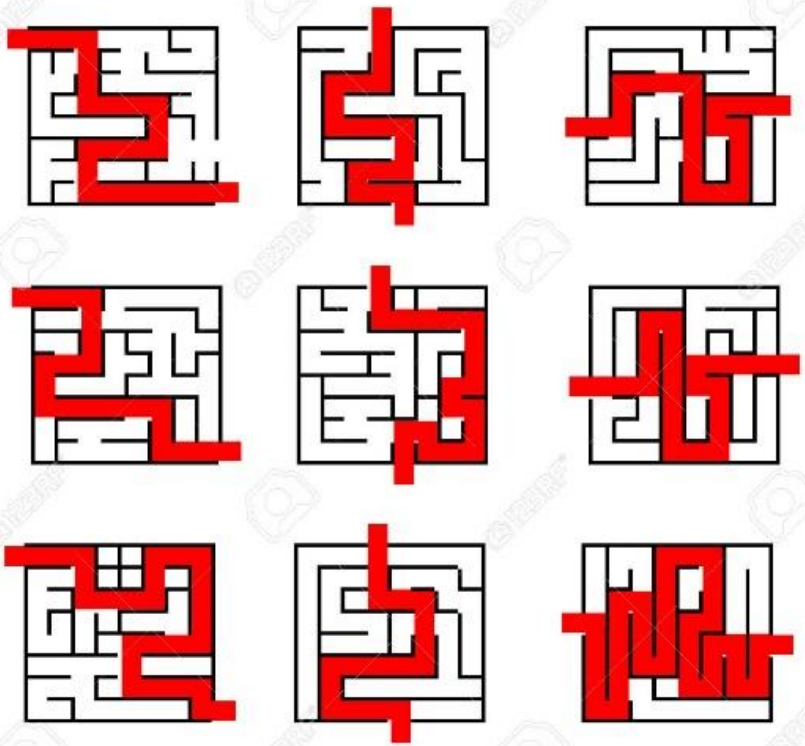
① Memory를 가지는 Model : RNN의 Hidden State 이용하여 다양한 task에 기억하며 학습 가능

② Meta Learning 알고리즘 : test 시점에 경험하지 못한 task를 빠르게 해결하기 위한 목적으로 model weight 를 어떻게 update할 것인가?

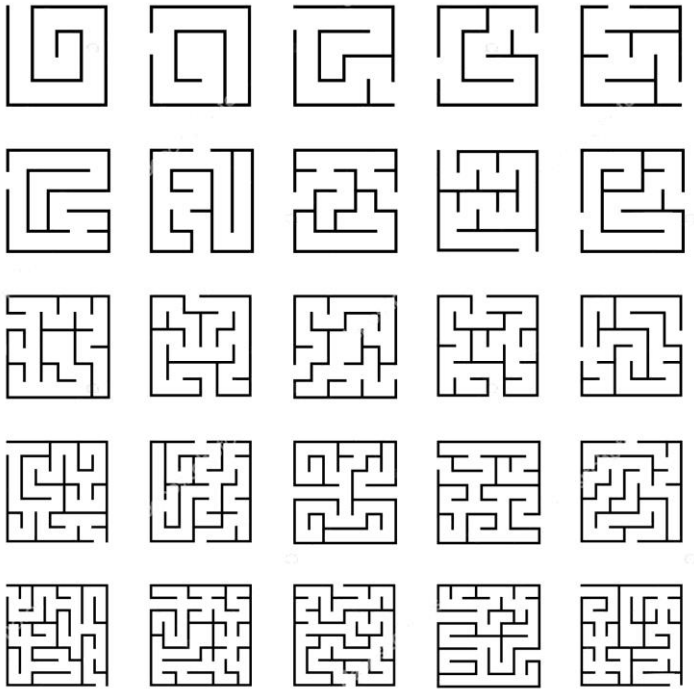
③ 다양한 MDP : Agent가 학습하는 동안 다양한 환경을 접했을 때 서로 다른 MDP에 대해서 어떻게 학습할 것인가?

Meta Reinforcement Learning

메타 강화학습을 통해 우리는 다음을 기대할 수 있다.



같은 환경에서 다양한 플레이



경험하지 못한 환경에서 좋은 성능

Meta Reinforcement Learning

메타 강화학습은 아래와 같이 다양한 접근 방법이 있다.

RL^2(2016, OpenAI), Learning to reinforcement learn(2017, Google)

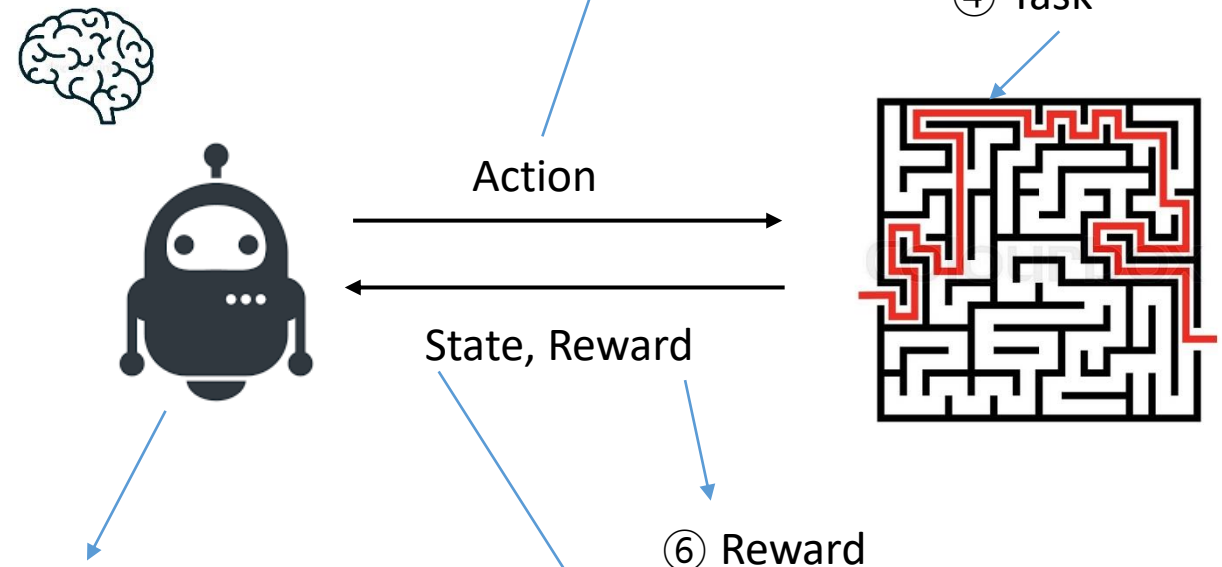
① Policy

Meta-Reinforcement Learning of Structured Exploration Strategies(Gupta et al, UC Berkeley, ICML 2018)

② Action

Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and(2019, Uber)

④ Task



⑥ Reward

Automatic Goal Generation for Reinforcement Learning Agents(Carlos et al, ICML 2018)

⑤ State

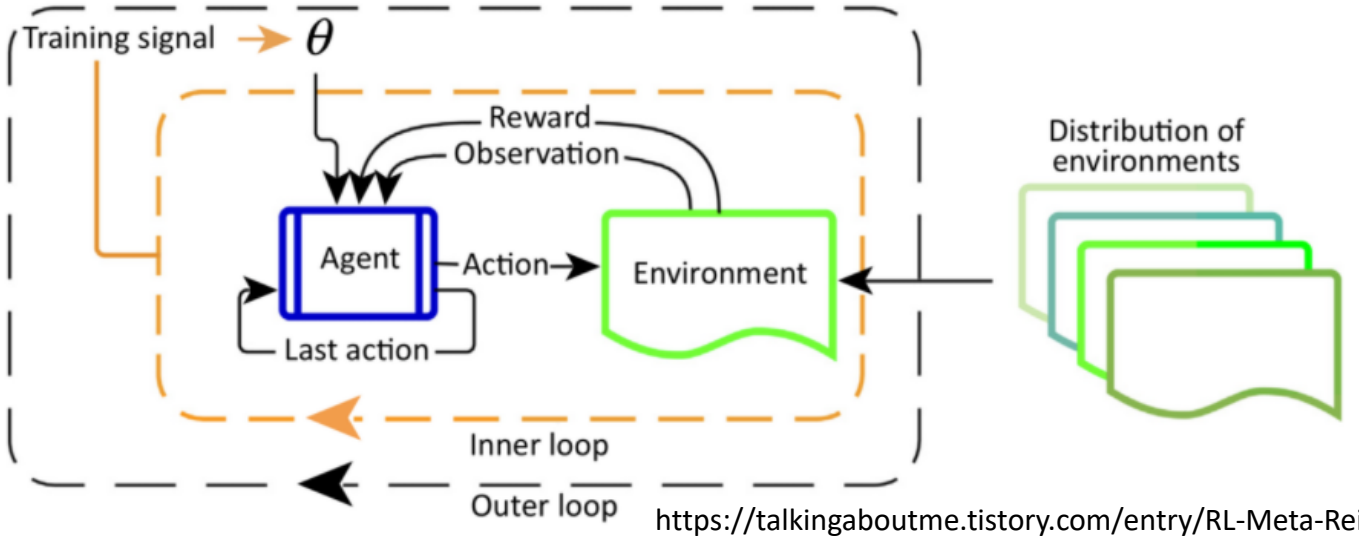
Automatic Data Augmentation for Generalization in Deep reinforcement Learning (NYU, Facebook AI , 2021)

③ Agent Design

Reinforcement Learning for Improving Agent Design(Ha, ICML 2018)

RL^2 , Learning to reinforcement learn

Train task와 Test task가 달라지는 경우 서로 다른 Reward Probability와 State Transition Probabilities에 대해서도 학습을 하는 방법



https://talkingaboutme.tistory.com/entry/RL-Meta-Reinforcement-Learning

$P_{ss'}^{a}$, $R(s,s',a)$의 변화 또한 학습하기 위해 LSTM
Policy라는 별도의 Outer loop 학습 (Slow Learning)

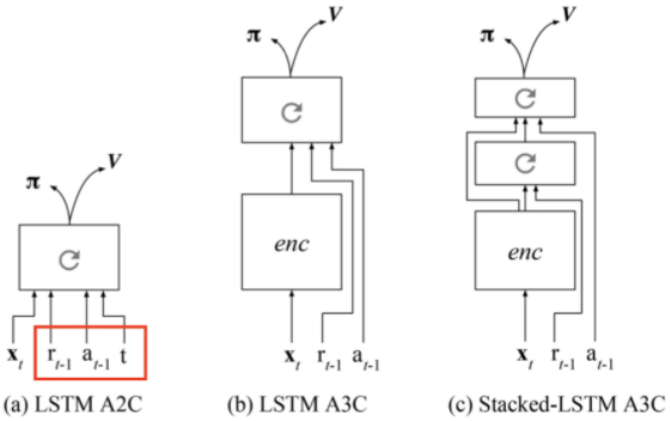
RL^2: **Fast** REINFORCEMENT LEARNING VIA **SLOW** REINFORCEMENT LEARNING

RL^2: Fast REINFORCEMENT LEARNING VIA SLOW REINFORCEMENT LEARNING (2016, OpenAI)
Learning to reinforcement learn(2017, Google)

RL^2 , Learning to reinforcement learn

Policy | Action | Agent | Task | State | Reward

정책 네트워크에 현 상태 외, 이전 액션과 리워드를 같이 넣어주어 환경 자체도 학습



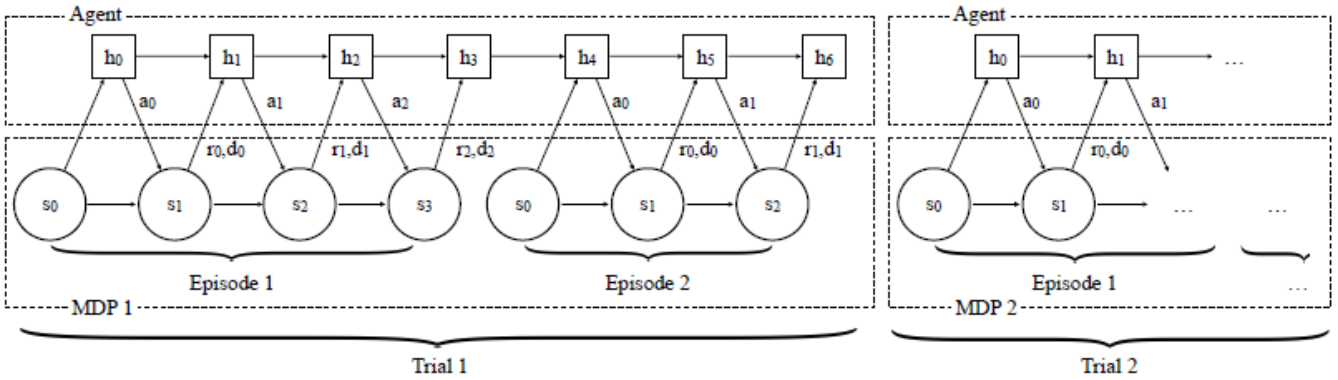


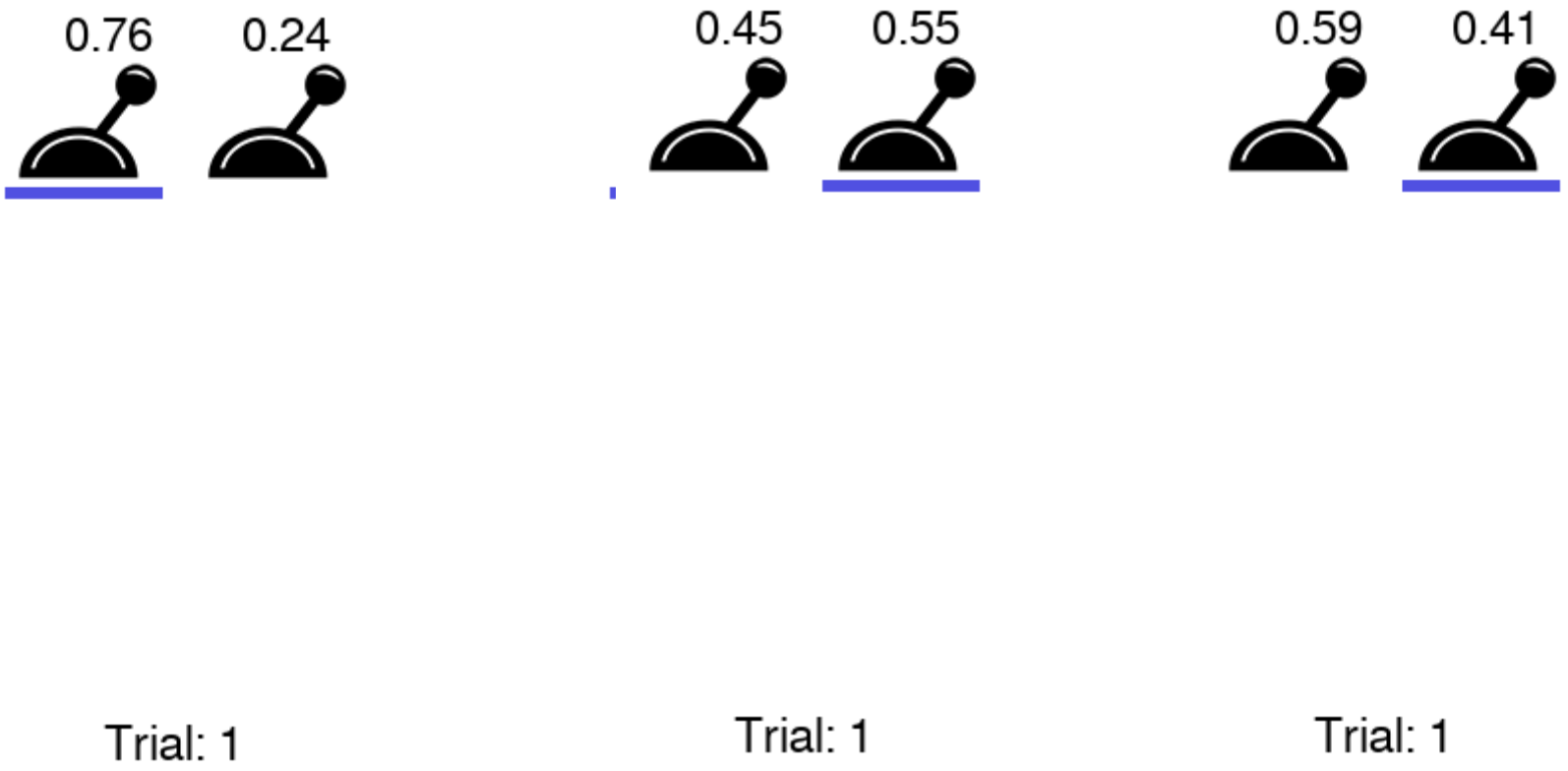
Figure 1: Procedure of agent-environment interaction

RL^2: Fast REINFORCEMENT LEARNING VIA SLOW REINFORCEMENT LEARNING (2016, OpenAI)
Learning to reinforcement learn(2017, Google)

DMQA

RL^2 , Learning to reinforcement learn

Policy | Action | Agent | Task | State | Reward

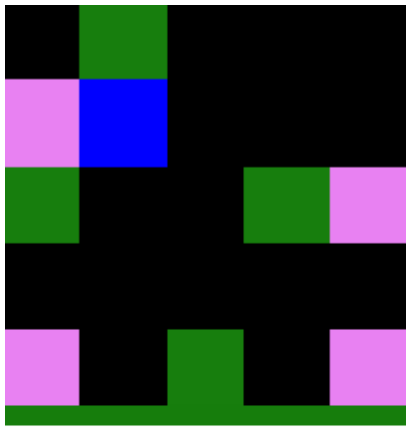


RL^2: Fast REINFORCEMENT LEARNING VIA SLOW REINFORCEMENT LEARNING (2016, OpenAI)
Learning to reinforcement learn(2017, Google)

DMQA

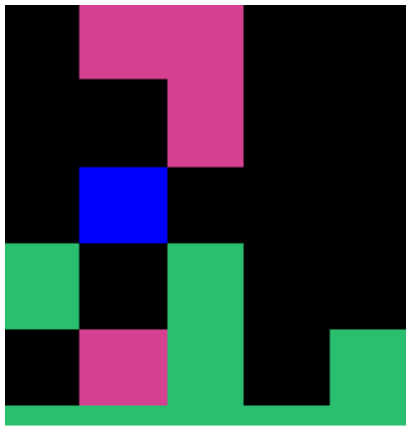
RL^2 , Learning to reinforcement learn

Policy | Action | Agent | Task | State | Reward



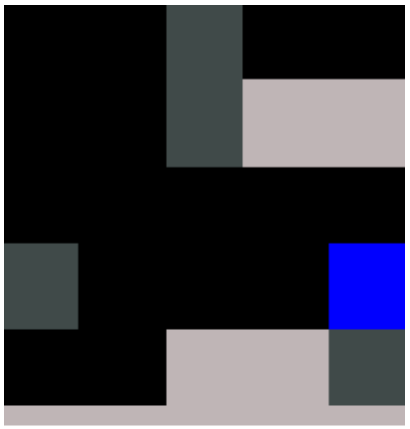
Step: 0

Reward: 0.0



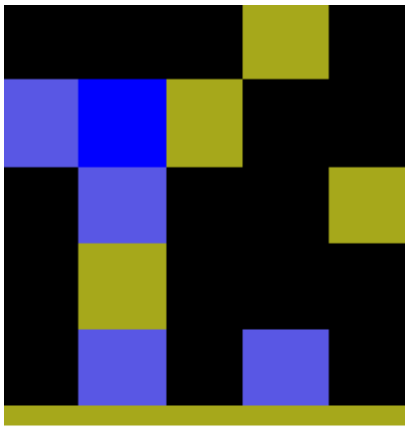
Step: 0

Reward: 0.0



Step: 0

Reward: 0.0

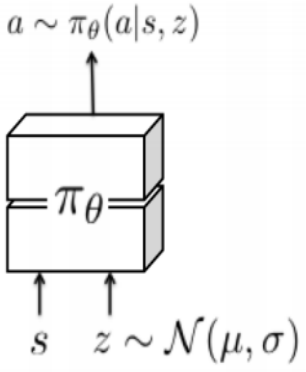


Step: 0

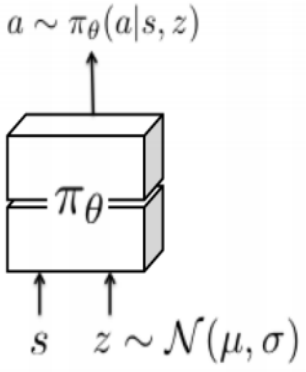
Reward: 1.0

RL^2: Fast REINFORCEMENT LEARNING VIA SLOW REINFORCEMENT LEARNING (2016, OpenAI)
Learning to reinforcement learn(2017, Google)

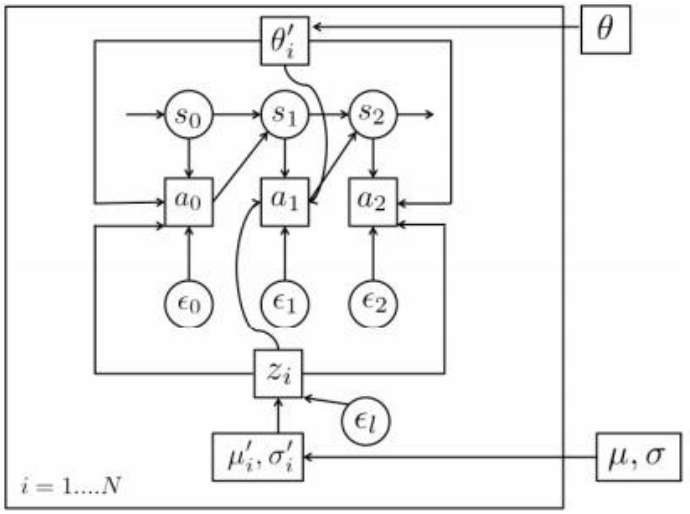
각 주어진 태스크에 맞는 Exploration을 각자 부여하여 다양한 Task에 좋은 성능을 냄



One Task

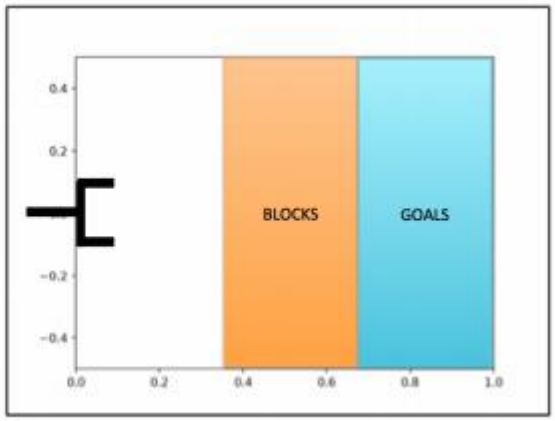


Other Task

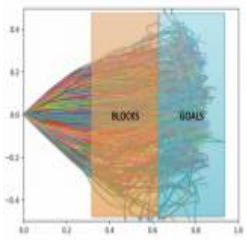


Meta-Reinforcement Learning of Structured Exploration Strategies(Gupta et al, UC Berkeley, ICML 2018)

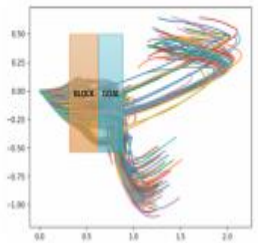
Meta-Reinforcement Learning of Structured Exploration Strategies



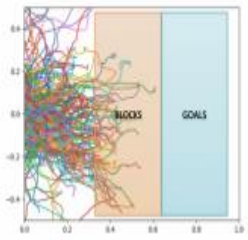
MAESN

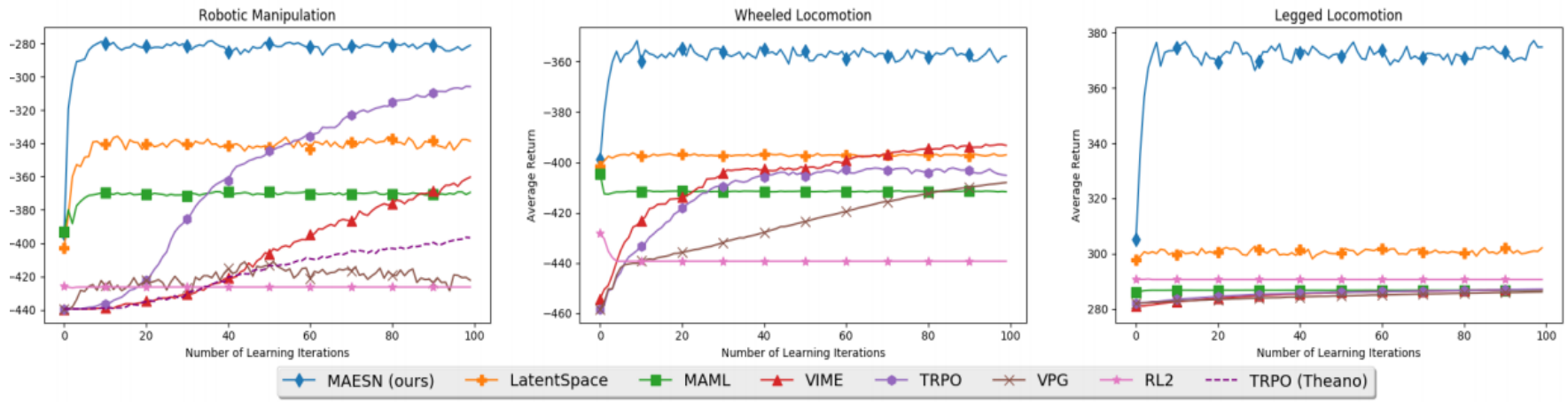


MAML



Random

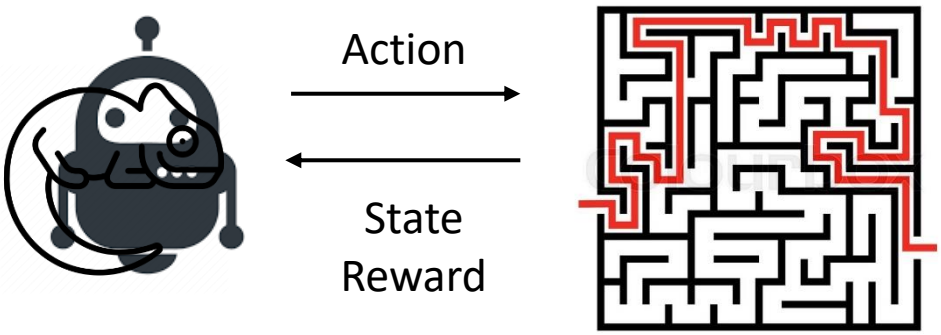




Meta-Reinforcement Learning of Structured Exploration Strategies(Gupta et al, UC Berkeley, ICML 2018)

Reinforcement Learning for Improving Agent Design

정책이 학습시에 환경의 상태 정보 외에도 에이전트의 Body Structure 정보도 학습하여 상황에 맞게 에이전트를 디자인



Reinforcement Learning for Improving Agent Design(Ha, ICML 2018)

에이전트의 Body Structure 특성(넓이, 길이, 무게) 변형하여 환경에 맞는 에이전트를 디자인하여 학습

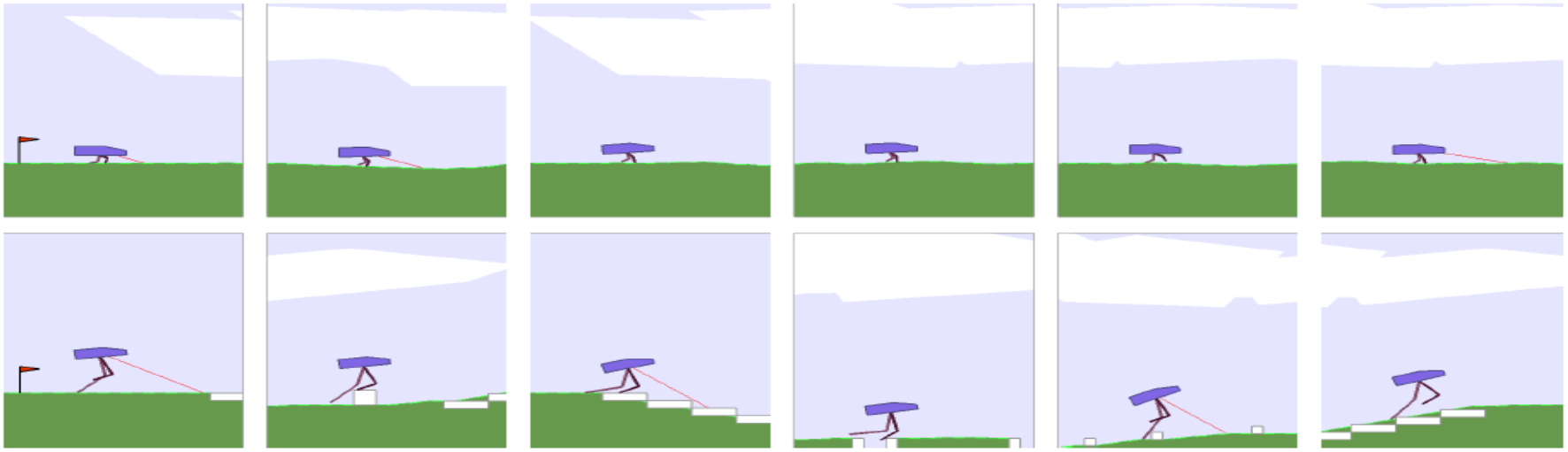
```
def rollout(agent, env):
  obs = env.reset()
  done = False
  cumulative_reward = 0
  while not done:
    a = agent.action(obs)
    obs, r, done = env.step(a)
    cumulative_reward += r
  return cumulative_reward
```

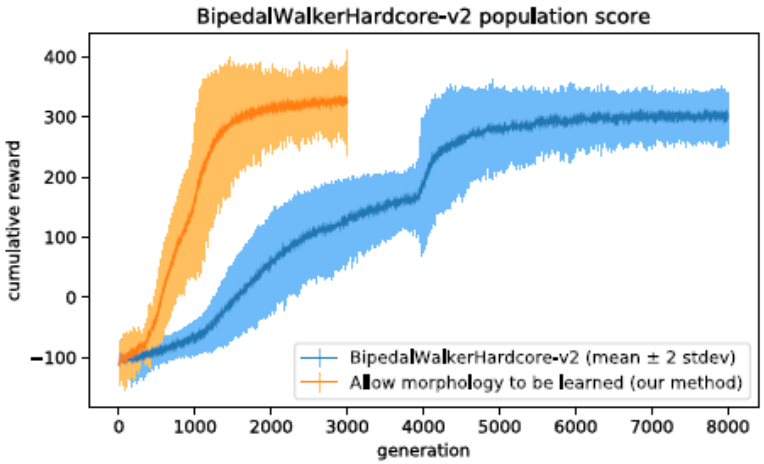
```
def rollout(agent, env_params, env):
  env.augment(env_params)
  obs = env.reset()
  done = False
  cumulative_reward = 0
  while not done:
    a = agent.action(obs)
    obs, r, done = env.step(a)
    r = augment_reward(r, env_params)
    cumulative_reward += r
  return cumulative_reward
```

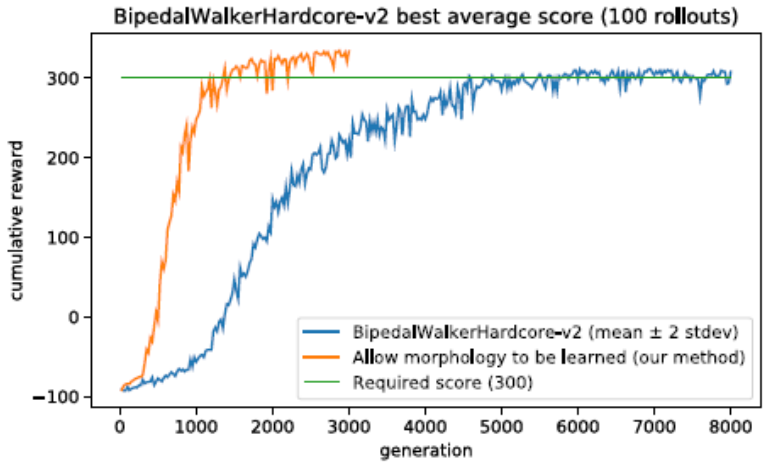
$$\pi_\theta(s,a) \rightarrow \pi_\theta(s,a,body\ parameter\ of\ agent)$$

Reinforcement Learning for Improving Agent Design(Ha, ICML 2018)

에이전트 변형이 없는 경우와 에이전트 변형을 가한 경우 비교 실험





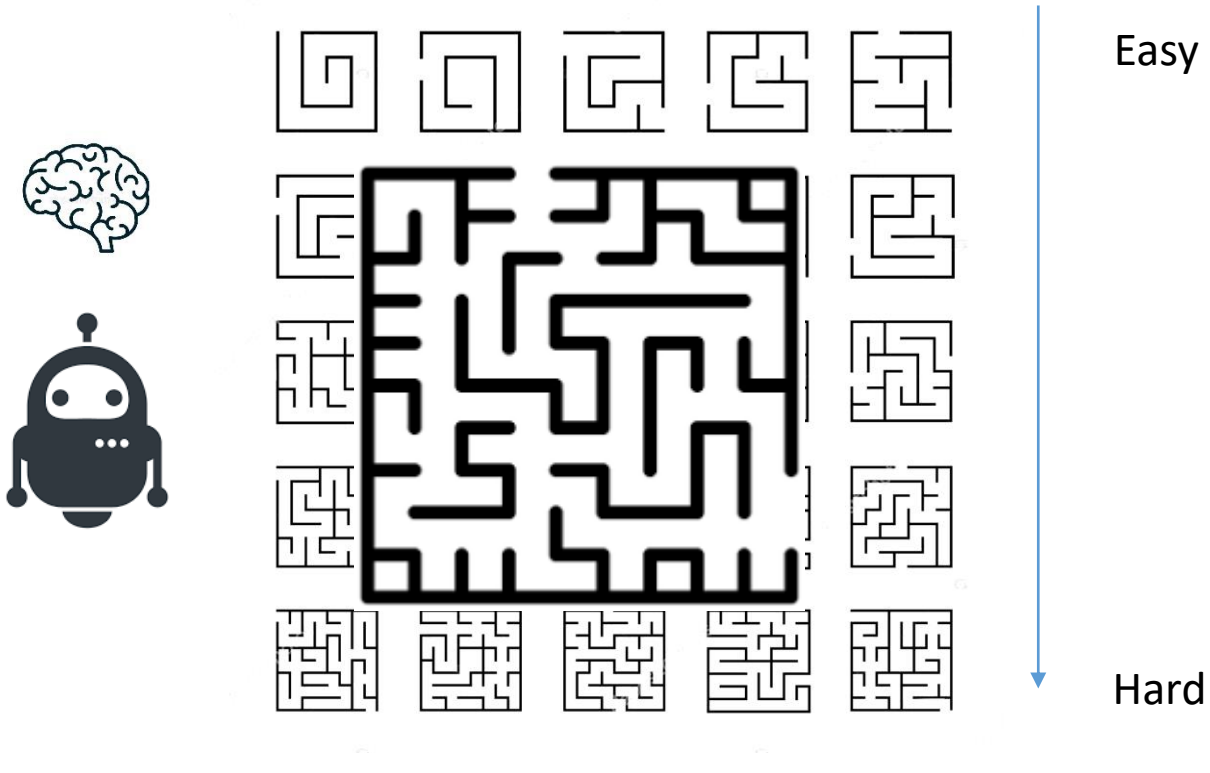


Reinforcement Learning for Improving Agent Design(Ha, ICML 2018)

Paired Open-Ended Trailblazer(POET)

메타 강화학습 3가지 핵심 요소 중 다양한 MDP를 어떻게 적절하게 학습할 것인가?

$$M_i =< \mathcal{S}, \mathcal{A}, \mathcal{P}_i, \mathcal{R}_i > \in \mathcal{M}$$

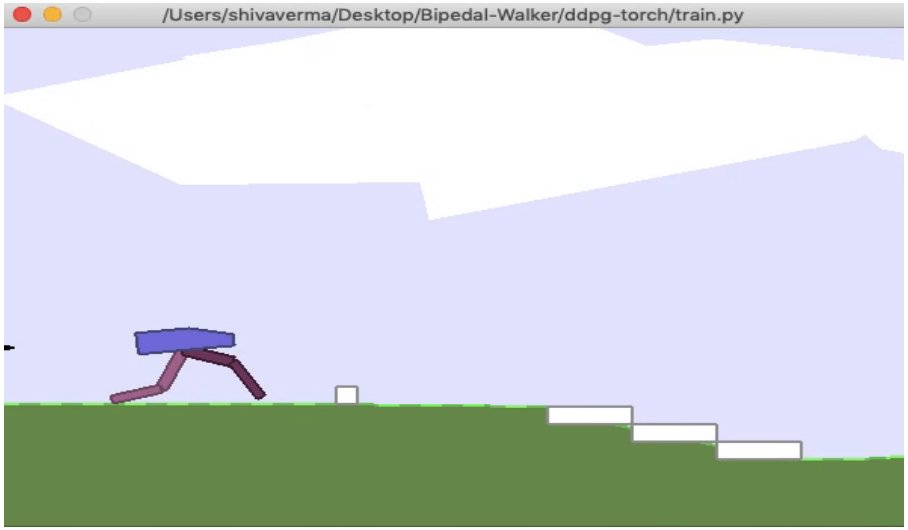


Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions (Rui et al, ICML 2020)

Paired Open-Ended Trailblazer(POET)

Policy | Action | Agent | **Task** | State | Reward

이 논문은 쉬운 환경에서 어려운 환경을 만들 수 있는 환경을 제공한다.



Bipedal Walker Hardcore

- Goal : 오른쪽 끝까지 이동
- Reward : Fall -100
- Maximum step : 2000

환경 난이도 조절 파라메터

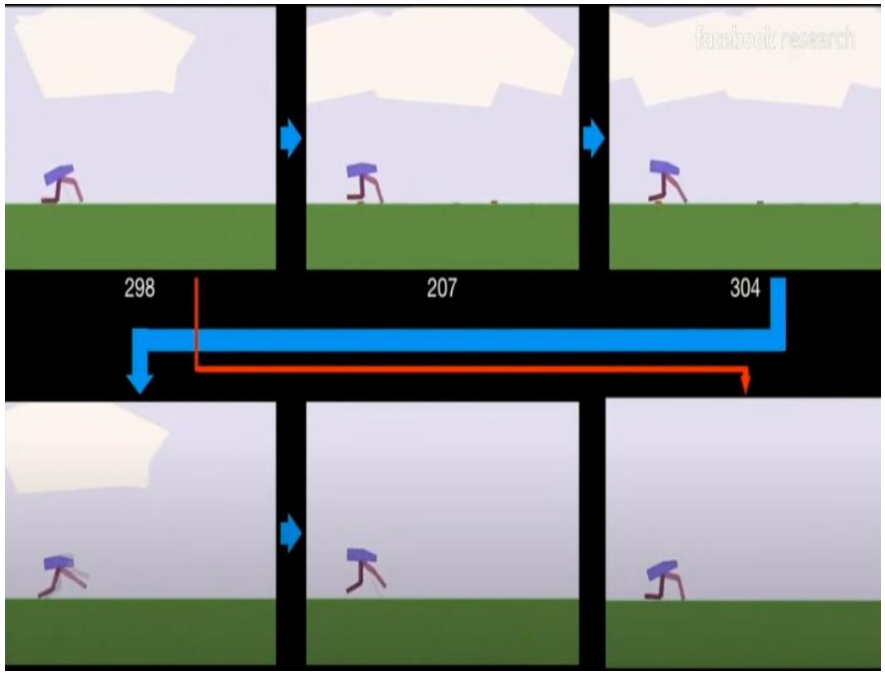
Obstacle Type	Stump Height	Gap Width	Step Height	Step Number	Roughness
Initial Value	(0.0, 0.4)	(0.0, 0.8)	(0.0, 0.4)	1	uniform(0, 0.6)
Mutation Step	0.2	0.4	0.2	1	uniform(0, 0.6)
Max Value	(5.0, 5.0)	(10.0, 10.0)	(5.0, 5.0)	9	10.0

Table 1: Environmental parameters (genes) for the Bipedal Walker problem.

Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions (Rui et al, ICML 2020)

Paired Open-Ended Trailblazer(POET)

아래와 같이 에이전트는 점점 실력이 향상되어 진화하는 모습을 볼 수 있다.



Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions (Rui et al, ICML 2020)

Paired Open-Ended Trailblazer(POET)

에이전트는 다양한 환경에서 좋은 성능을 내는 것을 확인할 수 있다.

Roughness	0
Gap	0
Stump	0

Roughness	0.2
Gap	0 - 1.2
Stump	0.2 - 0.6

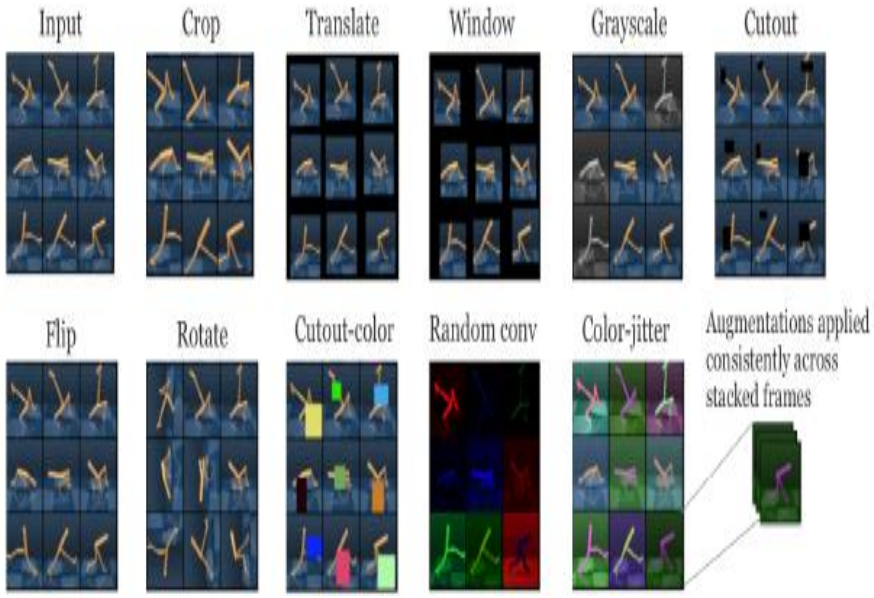
Roughness	2.2
Gap	0.8-1.2
Stump	0.2-0.4

Easy, Simple

Challenging , Complex

Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions (Rui et al, ICML 2020)

State Augmentation + Data Regularization



Reinforcement Learning with Augmented Data(Laskin et al, UC Berkely, ICML 2020)

$$f: \mathcal{S} \times \mathcal{H} \rightarrow \mathcal{S} \ (Image\ Transformation)$$

$$J_{PG}(\theta) = \sum_{a \in A} \pi_\theta(a|s) \hat{A}_{\theta_{old}}(s,a)$$

$$= \mathbb{E}_{a \sim \pi_{\theta_{old}}} \left[\frac{\pi_\theta(a|f(s))}{\pi_{\theta_{old}}(a|s)} \right] \hat{A}_{\theta_{old}}(s,a)$$

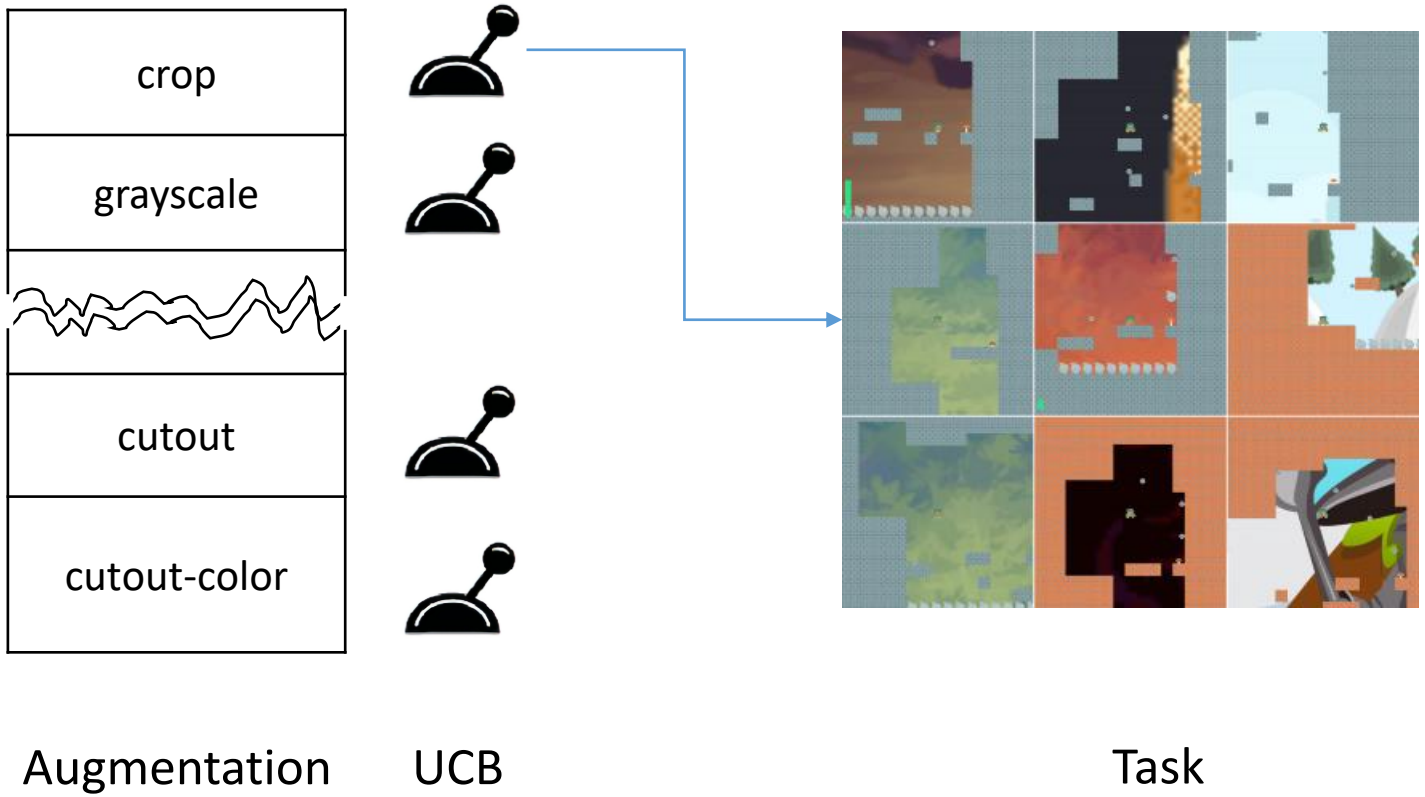
$$J_{DRAC} = J_{PPO} - \alpha_r (G_\pi + G_v)$$

$$G_\pi = KL[\pi_\theta(a|f(s,v)) | \pi(a|s)]$$

$$G_v = \left(V_\theta(f(s,v)) - V(s) \right)^2$$

Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels(Yarts et al, Facebook AI, ICLR 2021)

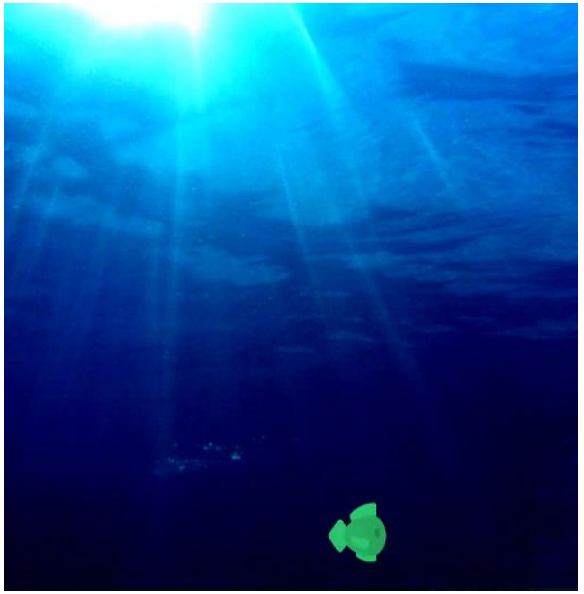
State Augmentation + Data Regularization + **밴딧 알고리즘(UCB)**

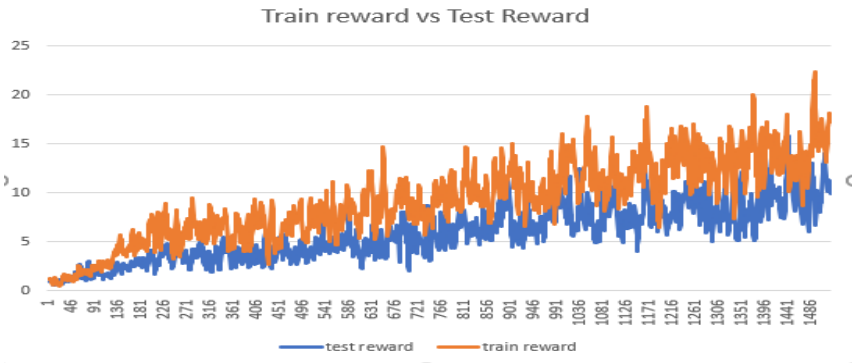


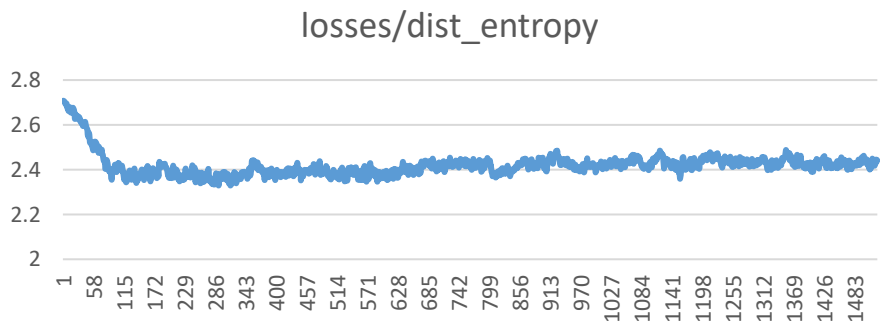
Automatic Data Augmentation for Generalization in Deep reinforcement Learning (NYU, Facebook AI , 2021)

Automatic Data Augmentation for Generalization in Deep reinforcement Learning

에이전트는 이미지 정보로 학습되나 배경이 달라져도 좋은 성능이 보장된다.







Automatic Data Augmentation for Generalization in Deep reinforcement Learning (NYU, Facebook AI , 2021)

논문에서 가정한데로, 각 게임마다 잘되는 Augmentation 기법들이 각각 있다.

Game	BigFish	StarPilot	FruitBot	BossFight	Ninja	Plunder	CaveFlyer	CoinRun
Best Augmentation	crop	crop	crop	flip	color-jitter	crop	rotate	random-conv

Game	Jumper	Chaser	Climber	Dodgeball	Heist	Leaper	Maze	Miner
Best Augmentation	random-conv	crop	color-jitter	crop	crop	crop	crop	color-jitter

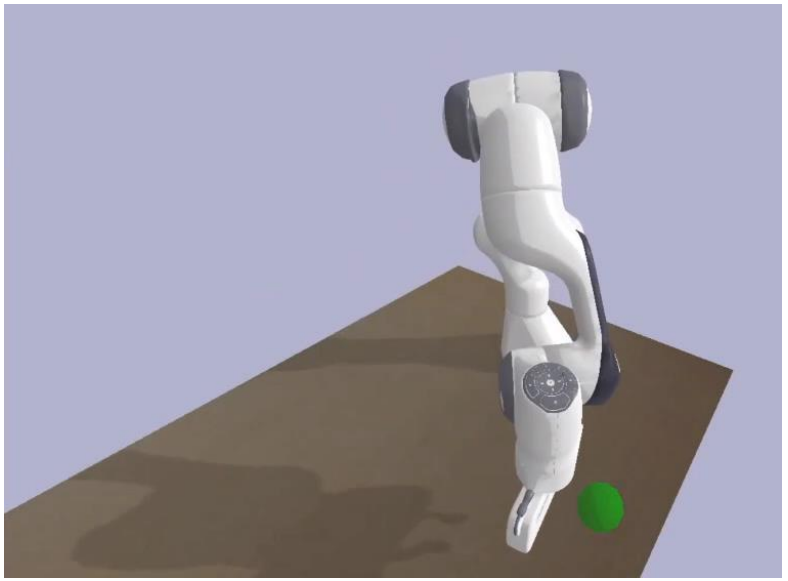
Automatic Data Augmentation for Generalization in Deep reinforcement Learning (NYU, Facebook AI , 2021)

Automatic Goal Generation for Reinforcement Learning Agents

기존 리워드로 학습하는 대신에 에이전트를 다양한 태스크마다 다른 Goal을 부여하며, 이를 GAN(Goal GAN)으로 학습

Algorithm 1 Generative Goal Learning

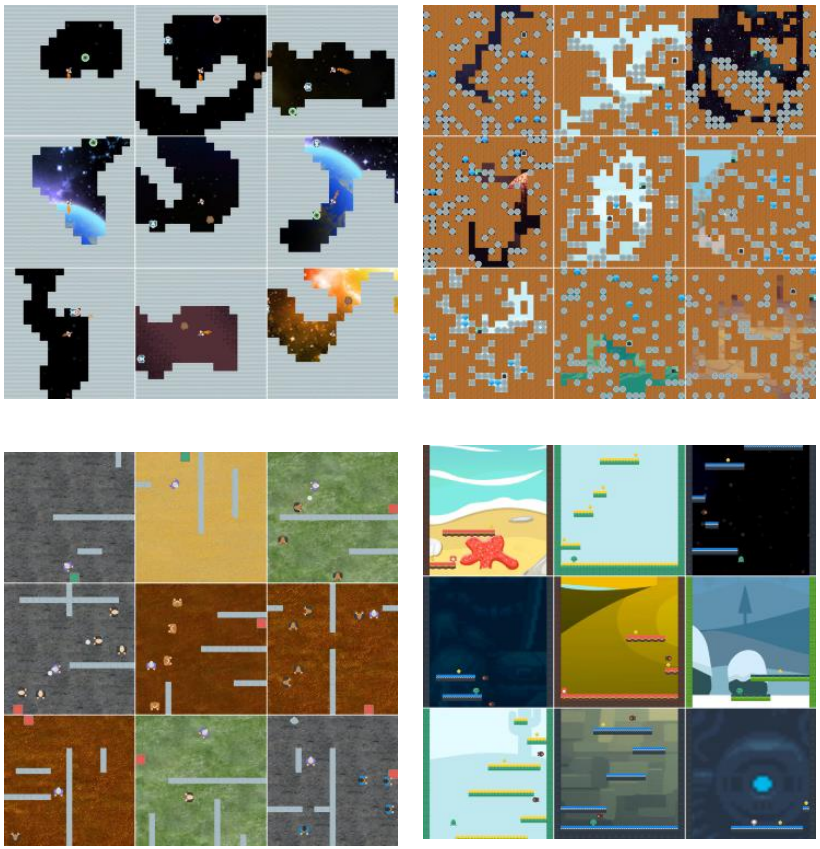
Input: Policy π_0
Output: Policy π_N
$(G, D) \leftarrow$ `initialize_GAN()`
$goals_{\text{old}} \leftarrow \varnothing$
for $i \leftarrow 1$ **to** N **do**
  $z \leftarrow$ `sample_noise`$(p_z(\cdot))$
  $goals \leftarrow G(z) \cup$ `sample`$(goals_{\text{old}})$
  $\pi_i \leftarrow$ `update_policy`$(goals, \pi_{i-1})$
  $returns \leftarrow$ `evaluate_policy`$(goals, \pi_i)$
  $labels \leftarrow$ `label_goals`$(returns)$
  $(G, D) \leftarrow$ `train_GAN`$(goals, labels, G, D)$
  $goals_{\text{old}} \leftarrow$ `update_replay`$(goals)$
end for



Automatic Goal Generation for Reinforcement Learning Agents(Carlos et al, ICML 2018)

Environment for Assessment

OpenAI 에서 제공하는 환경으로 16개의 게임들이 제공된다.



Procgen Benchmark(OpenAI)

- **High Diversity**: The diversity in the resulting level distributions presents agents with meaningful generalization challenges.

- **Fast Evaluation**: 200M timesteps. perform thousands of steps per second on a single CPU core, enabling a fast experiment

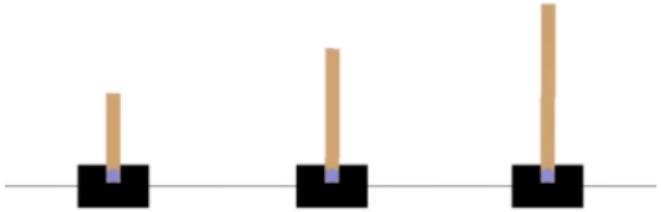
- **Tunable Difficulty**: All environments support two well-calibrated difficulty settings: easy and hard.

https://openai.com/blog/procgen-benchmark/

Environment for Assessment

UC Berkeley에서 제공하는 환경으로 5개의 OpenAI Gym Variants 들이 제공된다.

Environment	Parameter	D	R	E
CartPole	Force	10	[5,15]	[1,5] U [15,20]
	Length	0.5	[0.25,0.75]	[0.05,0.25] U [0.75,1.0]
	Mass	0.1	[0.05,0.5]	[0.01,0.05] U [0.5,1.0]
MountainCar	Force	0.001	[0.0005,0.005]	[0.0001,0.0005] U [0.005,0.01]
	Mass	0.0025	[0.001,0.005]	[0.0005,0.001] U [0.005,0.01]
Acrobot	Length	1	[0.75,1.25]	[0.5,0.75] U [1.25,1.5]
	Mass	1	[0.75,1.25]	[0.5,0.75] U [1.25,1.5]
	MOI	1	[0.75,1.25]	[0.5,0.75] U [1.25,1.5]
Pendulum	Length	1	[0.75,1.25]	[0.5,0.75] U [1.25,1.5]
	Mass	1	[0.75,1.25]	[0.5,0.75] U [1.25,1.5]
HalfCheetah	Power	0.90	[0.70,1.10]	[0.50,0.70] U [1.10,1.30]
	Density	1000	[750,1250]	[500,750] U [1250,1500]
	Friction	0.8	[0.5,1.1]	[0.2,0.5] U [1.1,1.4]
Hopper	Power	0.75	[0.60,0.90]	[0.40,0.60] U [0.90,1.10]
	Density	1000	[750,1250]	[500,750] U [1250,1500]
	Friction	0.8	[0.5,1.1]	[0.2,0.5] U [1.1,1.4]



Deterministic (D): The parameters of the environment are fixed at the default values in the implementations from Gym and Roboschool. That is, every time the environment is reset, only the state is reset.

Random (R): Every time the environment is reset, the parameters are uniformly sampled from a d-dimensional box containing the default values. This is done by independently sampling each parameter uniformly from an interval containing the default value.

Extreme (E): Every time the environment is reset, its parameters are uniformly sampled from 2^d d-dimensional boxes anchored at the vertices of the box in R. This is done by independently sampling each parameter uniformly from the union of two intervals that straddle the corresponding interval in R.

Assessing Generalization in Deep Reinforcement Learning(Packer et al, UC Berkeley, 2019)

Conclusions

"복잡한 세상을 모델링 할 때 우리의 Domain Knowledge나 Handcrafted code들을 AI 안에 집어넣는 것은 발견의 과정이 어떻게 이루어지는지 이해하는 것을 더 어렵게 만들 뿐이다."

Bitter Lessons by Richard Sutton