



AI 프렌즈 Hot Issue(Short Talk): Model based Reinforcement Learning

ETRI, Kim Kwihoon
(kwihoon@etri.re.kr)



발표는 이런 식으로,,



1

RL overview & RL에 주목하는 이유?

2

RL Tech. Tree

3

Model-based RL vs Model-free RL

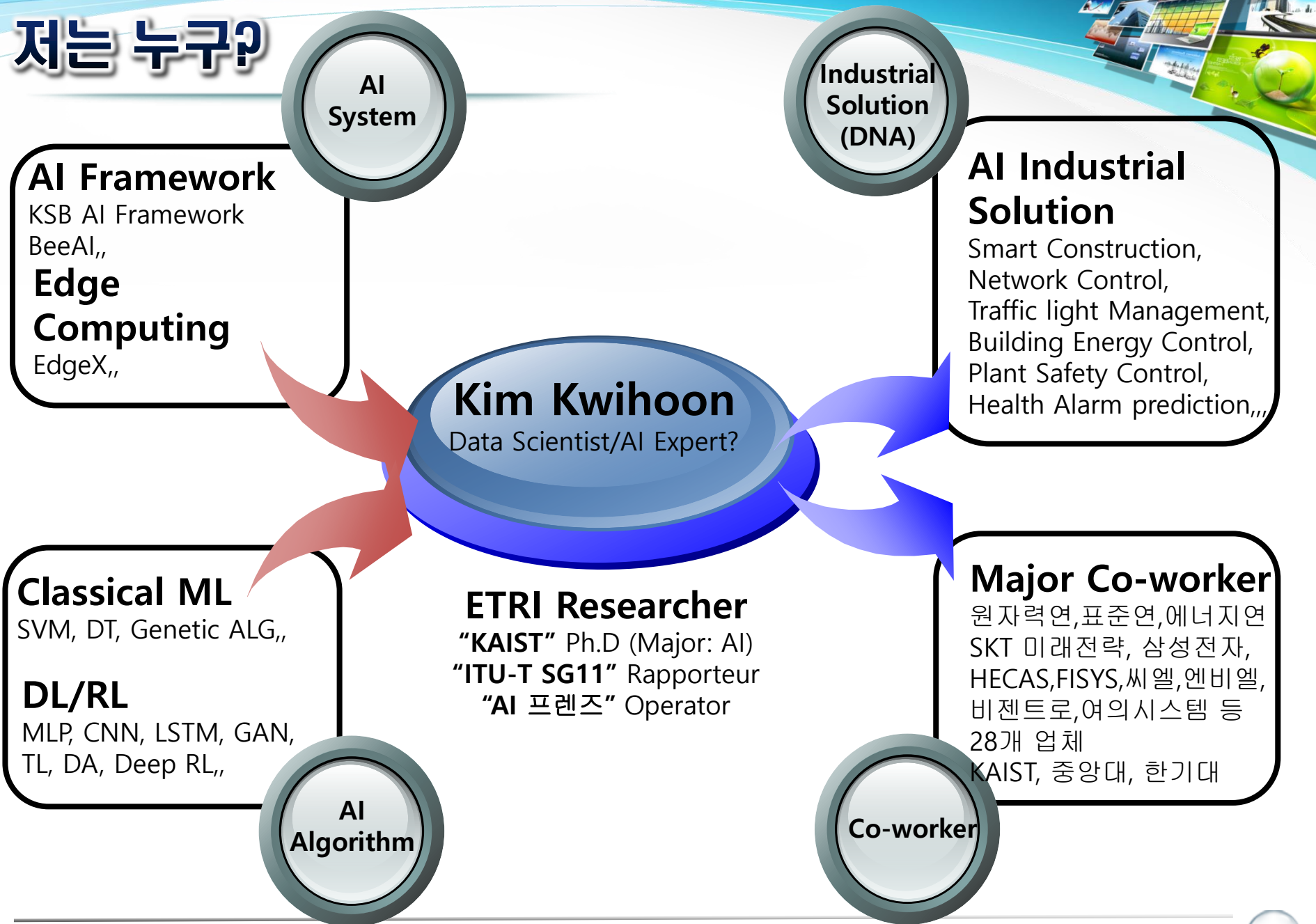
4

몇 가지 사례들

5

Summary

저는 누구?



1.1 RL Overview

- Labeled data
- Direct feedback
- Predict outcome/future



- No labels
- No feedback
- "Find hidden structure"

- Decision process
- Reward system
- Learn series of actions

■ "Pure" Reinforcement Learning (cherry)

- ▶ The machine predicts a scalar reward given once in a while.
- ▶ **A few bits for some samples**

■ Supervised Learning (icing)

- ▶ The machine predicts a category or a few numbers for each input
- ▶ Predicting human-supplied data
- ▶ **10→10,000 bits per sample**

■ Unsupervised/Predictive Learning (cake)

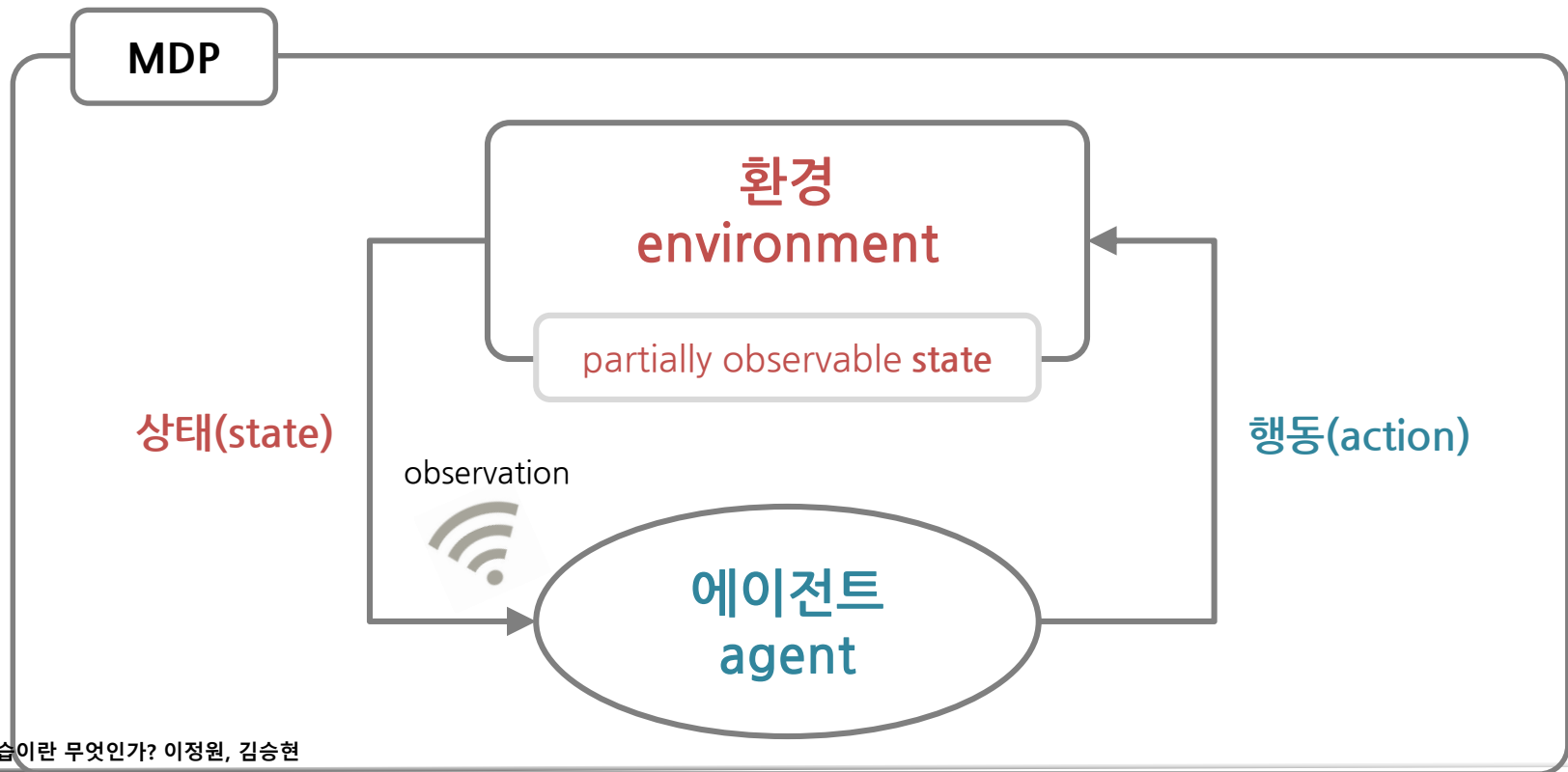
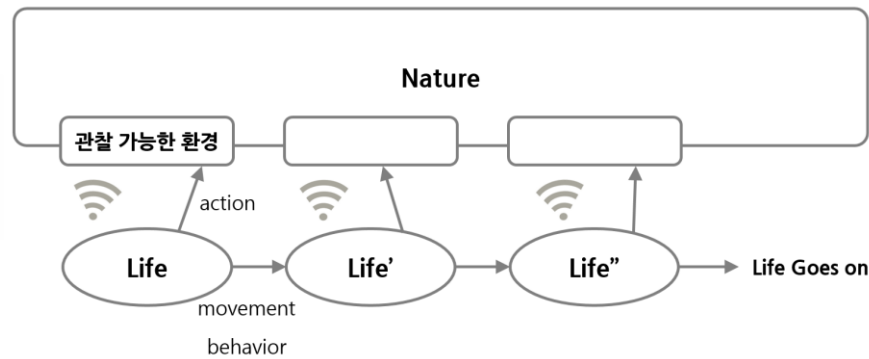
- ▶ The machine predicts any part of its input for any observed part.
- ▶ Predicts future frames in videos
- ▶ **Millions of bits per sample**



■ (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

From Yann Lecun, (NIPS 2016)

1.1 RL Overview



* 강화학습이란 무엇인가? 이정원, 김승현

1.2 RL에 주목하는 이유?



1. 진정한 인공지능의 가능성

강화학습은 지도학습과는 달리 에이전트가 존재합니다. 에이전트는 주어진 환경에서 스스로 행동을 선택하며 학습에 필요한 데이터를 모읍니다. 학습을 하기 위해 새로운 정보가 필요하다면 에이전트는 exploration을 할 것입니다. 이미 충분히 환경을 탐험했다면 주어진 데이터에 대해서 exploitation을 할 것입니다. 사람도 새로운 환경에서 무엇인가를 배울 때 스스로 데이터를 수집하면서 학습합니다. 따라서 강화학습은 지도학습보다는 조금 더 사람의 학습 방법에 가깝다고 말할 수 있습니다.

강화학습의 중요한 특징 중 하나는 경험을 통해 학습한다는 것입니다. Trial & error라고도 부르는 이 방법은 직접 시도를 한다는 것이 독특합니다. 강화학습 에이전트가 시도를 통해 혹은 경험을 통해 학습하기 때문에 비효율적이라는 단점이 있습니다. 하지만 이런 특성은 동물의 자연스러운 학습 방법을 닮아있습니다. 강화학습이 현재는 당장 비효율적인 면이 많지만 앞으로 발전할 수 있는 면도 많습니다. RLKorea 운영진은 미래에 투자한다는 개념으로 강화학습을 공부하고 있습니다.

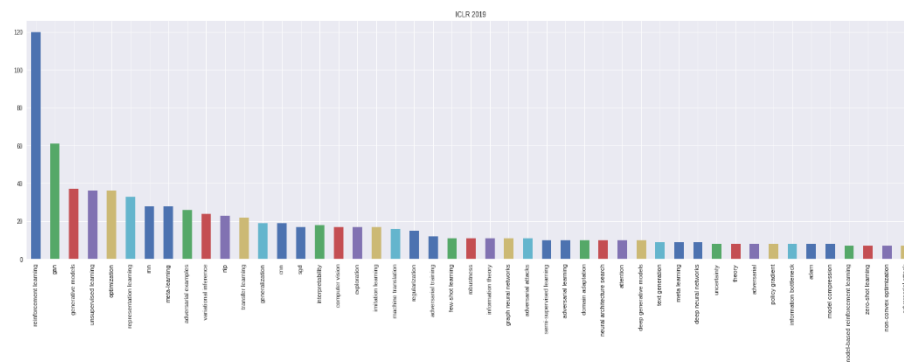
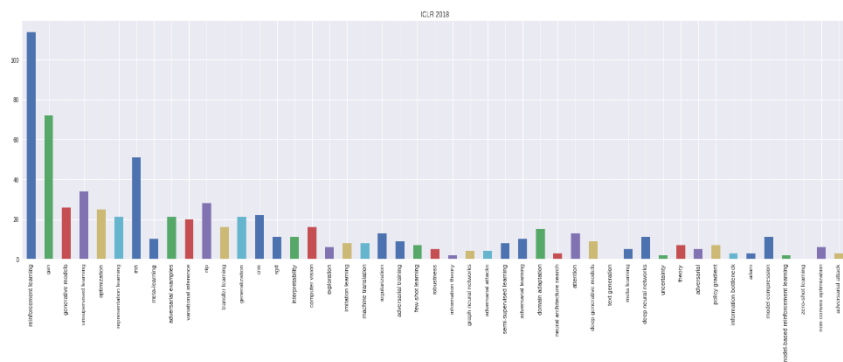
* RLKorea 운영진의 의견

1.2 RL에 주목하는 이유?



2. 강화학습 논문의 양 증가

딥러닝 분야에서 유명한 국제 학회는 ICLR, NIPS, ICML이 있습니다. 최근에 Harvard NLP에서 2018년, 2019년 ICLR에 제출한 논문을 분석했습니다. 아래 첫번째 그림은 2018년 ICLR에 제출된 논문이고 분야별로 몇 개의 논문이 제출되었는지를 보여줍니다. 무수히 많은 논문이 쏟아지는 GAN 보다도 더 많은 논문이 제출된 것을 볼 수 있습니다. 두번째 그림은 2019년 ICLR에 제출된 논문을 보여줍니다. 흥미로운 점은 **GAN의 논문 수는 줄어든 반면 강화학습의 논문 수는 늘었다**는 것입니다.



* RLKorea 운영진의 의견

1.2 RL에 주목하는 이유?



3. 인공지능 선도 기업의 활발한 연구

현재 인공지능 분야는 거대한 IT 기업들이 선도해가고 있습니다. 구글, 페이스북, 마이크로소프트, OpenAI에서 많은 딥러닝 논문을 출판하고 있습니다. 특히 DeepMind와 OpenAI에서 수많은 강화학습 논문을 내고 있습니다. FAIR(Facebook AI Research)와 Microsoft Research에서도 꾸준히 강화학습 논문을 내고 있습니다. 학계가 아닌 기업에서 활발히 강화학습을 한다는 사실을 보면 강화학습은 미래가 밝은 기술이라고 생각할 수 있습니다. 또한 점점 다양한 application에 적용한 논문이 나오는 것을 보면 강화학습이 적용되는 범위가 넓어지고 있다는 것을 알 수 있습니다.

* RLKorea 운영진의 의견

1.2 RL에 주목하는 이유?



딥마인드

위키백과, 우리 모두의 백과사전.

딥마인드(**영어**: DeepMind Technologies Limited)는 **알파벳**의 자회사이자 **영국**의 **인공지능**(AI) 프로그램 개발 회사이다.

13세에 세계 유소년 **체스** 대회 2위를 한 천재 **데미스 허사비스**가 15세 때 고교과정을 마치고 케임브리지대에서 컴퓨터공학 학사, 유니버시티칼리지런던(UCL)에서 인지신경과학 박사 학위를 받으며 2010년 **신경과학을 응용한 인공지능 회사**를 세운게 시조이다.^[1]

장소는 **영국**의 **런던**이었으며 **데미스 허사비스** 외에도 셰인 레그, 무스타파 술레이만 등이 공동 창업하였다. 당시 회사 이름은 '딥마인드 테크놀로지'였다. 머신러닝(기계학습)과 신경과학 기반의 스스로 학습하는 컴퓨터 알고리즘을 개발한다. 미리 프로그램이 짜여져 있는 기존 인공지능과 달리 머신러닝으로 스스로 정보를 처리함으로써 특정 분야에 국한되지 않고 다양한 분야에서 활용할 수 있는 '범용 학습 알고리즘'을 만드는 것을 목표로 하고 있다. **구글**이 근래 인공지능 분야에 대한 투자를 확대하였고, 이 회사의 가치를 알아본 구글이 **2014년** 4억 달러(약 4800억원)에 인수해 현재의 사명이 되었다. 직원 규모는 2016년 100여명이다.^[2]

구글 딥마인드는 심층 인공지능 기술인 '심층 큐 네트워크'(Deep Q-network)를 독자적으로 개발하였다. 이 기술은 다층 신경망(Deep Neural Network)과 큐 러닝(Q-Learning)을 조합한 기술이다. 규칙을 알지 못하는 상태에서 점수와 픽셀 디스플레이를 정보로 활용하여 최고점을 만들기 위해 이전 게임 세션으로부터 학습하는 능력만을 갖추었다. 그리하여 아타리 2600 비디오 게임을 플레이하는 법을 스스로 터득했는데, 그 실력이 전문적인 게임 테스트의 실력에도 맞먹는 수준이었다고 한다. 해당 기술의 연구는 **네이처** 저널에 실렸다.^[3]

인공지능 바둑 프로그램인 **알파고**(AlphaGo)를 개발, 다른 바둑 프로그램들과 총 500회 대국을 벌여 499회 승리했다. 2015년 10월에는 바둑 기사 **판 후이**와 대국, 5전 전승하였다.(판 후이와 비공식 대결(시간 제한 30초) 3승 2패[알파고 승]). 2016년 3월 **이세돌** 9단과의 **알파고 대 이세돌** 대국에서 1회전과 2회전, 그리고 3회전에서 불계승하였으며 4회전에서는 **이세돌** 9단이 불계승 하였다. 마지막 5회전에서는 알파고의 승리로 끝이났다.^[4]

딥마인드 DeepMind Technologies Limited	
	
산업 분야	인공지능(AI)
창립	2010년 9월 23일 (8년 전)
창립자	데미스 허사비스 Demis Hassabis (CEO), 셰인 레그 Shane Legg, 무스타파 술레이만 Mustafa Suleyman
국가	영국
본사 소재지	영국 잉글랜드 런던
모기업	독립적 (2010년 ~ 2014년) 구글 (2014년 ~ 2016년) 알파벳 (2015년 ~ 현재)
웹사이트	공식 사이트 

스타크래프트 2 인공 지능 알파스타 [편집]

"저희는 DeepMind를 'AGI', 즉 인공 일반 지능이라고 부르는 **인공 지능**으로 만들고 있어요. 특정한 하나의 게임 에이전트만이 아니라 학습 패러다임을 이해함으로써 사전 지식 없이 어떤 게임이든 플레이할 수 있는 에이전트를 만드는 중이죠." - Oriol Vinyals

딥마인드의 **스타크래프트 2** 프로젝트 총책임자 Oriol Vinyals에 따르면 **스타크래프트 II**용으로 개발하고 있는 **인공 지능**은 스타크래프트 2뿐만이 아니라, 어떤 게임에도 적용할 수 있는 **인공 일반 지능**이 목표라고 한다.^[5]

구글 DeepMind의 스타크래프트 2 **인공 지능**은 **알파스타** (AlphaStar)로 명명되었다. **알파**에 **바둑**을 의미하는 Go를 붙여 명명된 **알파고** (AlphaGo)와 같은 명명 방식이다. 프로그래머 **다리오 윈치** (Dario Wunsch, TLO)와 **그레고리 코민츠** (Grzegorz Komincz, MaNa)를 상대로 각각 5:0으로 이겼다. 경기에서 AlphaStar의 평균 분당 행동수(APM)는 TLO와 MaNa보다 낮은 수준을 유지하였으나 최대 APM은 둘보다 높았다. 또한 시야가 모니터 크기로 제한되는 인간과 달리 맵 전체를 시야로 쓸 수 있었다. **시야 크기를 모니터 크기로 제한한 경기에서는** MaNa에게 0:1로 패배하였다.^[6]

1.2 RL에 주목하는 이유?



알파고의후예들: 알파고 Fan, Lee, Master, Zero

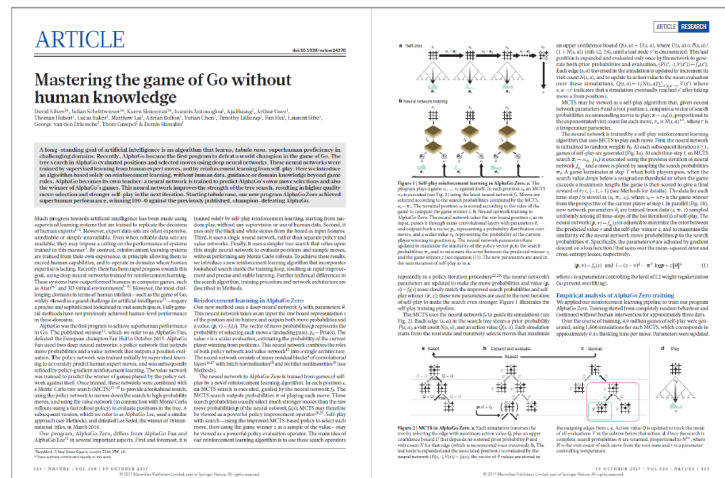
➤ 여러 버전의 알파고 등장

- 알파고 Fan
 - 2015년 10월 판 후이 2단을 이긴 버전
- 알파고 Lee
 - 2016년 3월 이세돌 9단을 이긴 버전
- 알파고 Master
 - 2017년 1월 온라인 대국 사이트에서 세계 최고 프로기사들을 60:0으로 제압
 - 커제 9단과의 대결에서 3:0 완승을 거뒀던 버전
- 알파고 Zero
 - 스스로 학습하여 신의 경지에 오른 버전 (2017.10.)

➔ 알파 Zero : 바둑 말고 다른 게임에 적용 가능한 알파 Zero (2018.01.)

➔ 알파 Fold : 단백질 3차 구조를 예측하는 알파폴드, 단백질 구조예측 학술대회(CASP) 우승 (2018.12.)

➔ 알파 스타 : 스타크래프트 대결 10:1 승리 (2019.01.)



1.2 RL에 주목하는 이유?



테리의 딥러닝 토크

1월 26일 오후 5:54 · 🌐

[스타크래프트2에서 인간을 이긴 DeepMind의 알파스타 분석]

알파고로 이세돌을 이겼던 구글 딥마인드가 스타크래프트2에서도 프로게이머를 이겼다. 최정상 게이머와 비교하면 중상위권에 해당한다고 한다. (예를 들면 ASL에 운이 좋아야 올라오는 정도..?) 누구는 이번 승리를 찬양하기도 하고, 누구는 이번 승리를 '비현실적 컨트롤의 승리'라며 깎아 내리는데, 무슨 말이 맞는지 몰라 테리가 세시간쯤 들여다봤다.

1. 스타와 바둑은 무엇이 다른가?

가장 큰 차이는 "모든 정보를 볼 수 있느냐 / 없느냐"에 차이가 있다. 바둑은 자신과 상대방이 똑같은 정보를 가지고 두뇌 싸움을 펼친다. 포커와 같이 숨겨진 패도 없고, 공공이라고 한다면 수들에 대한 계획, 즉 상대의 long-term planning을 알 수 없단 정도 일 것이다. 하지만 스타는 내 정보는 다 볼 수 있는 반면, 상대에 대한 정보는 두 유닛이 만나는 전장을 제외하곤 거의 볼 수 없다. 따라서 상대의 상황을 추측하며 맞춰가야 하는데, 그렇기에 좀더 추론해야 할 것이다.

또 크게 다른 점 중 하나는 real-time에 multi-agent problem 이라는 것이다. 하나의 주인공을 움직이는 것과 여러 유닛을 함께 움직여 하나의 목적을 이루는 건 난이도 자체가 하늘과 땅 차이이다. 게다가 유닛마다 종류도 다르고 속성도 달라 '조합'이란 걸 잘 맞춰야 한다. 그리고 real-time이기에 바둑처럼 몇분동안 생각할 수도 없다. 그리고 바둑은 내가 할 수 있는 action이 특정 위치에 돌을 놓는 것으로 한정되어 있지만, 스타는 이동, 공격, 마법 등의 여러가지 액션에, 위치도 19x19 바둑판에 제한되지 않는다.

이것만 봐도 스타가 얼마나 챌린징한 문제인지 이해할 수 있을 것이다. 예전에 체스보다 바둑이 훨씬 어렵다고 했는데, 그것은 주로 많은 경우의 수에 기인한 것이었다. 반면 스타는 새로운 챌린지들이 추가되었으니, imperfect information, real-time multi-agent planning, large action space, long-term planning 등이 그것이다.

1.2 RL에 주목하는 이유?



4. 각 도메인의 문제를 풀기 위해

딥러닝은 크게 Vision, NLP, Sound와 같은 도메인으로 나눌 수 있습니다. 강화학습은 게임, 제어, 자연어처리, 비전, 추천, 최적화로 나뉘볼 수 있습니다. 각 도메인에 속한 엔지니어는 그 도메인의 문제를 풀기 위해 다양한 기술을 사용합니다. 1, 2, 3에서 말한 이유가 아닌 현재 직면하고 있는 문제를 풀기 위해 강화학습을 공부해야 하는 경우가 있습니다. 각 도메인 별로 간단히 예시를 들어보겠습니다.

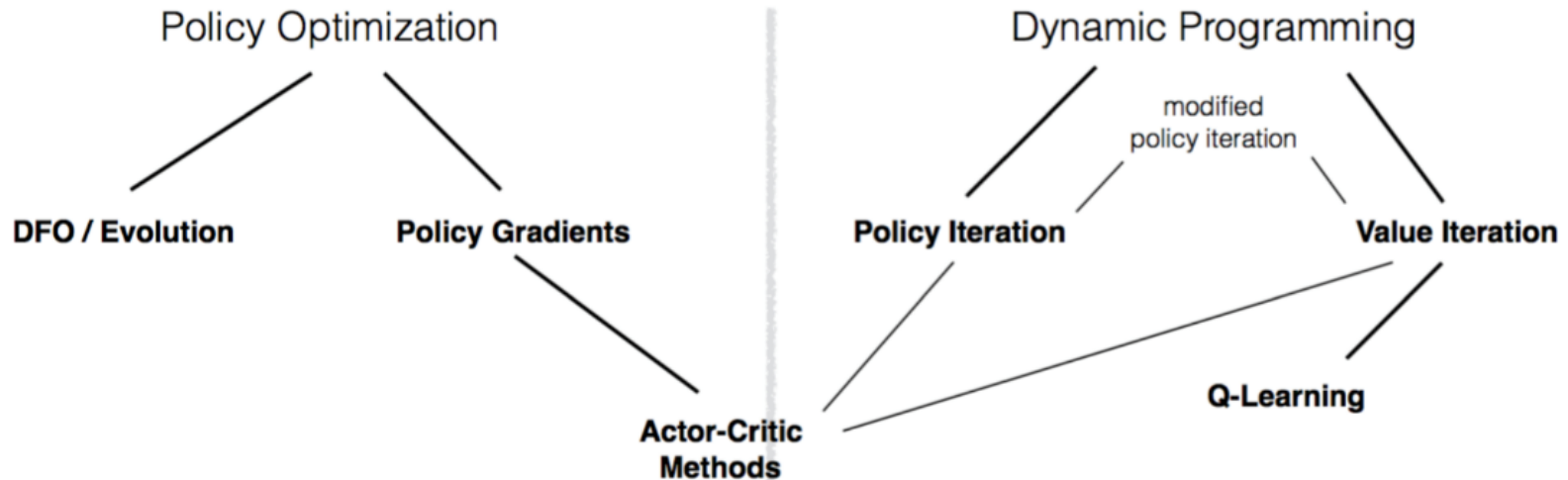
- | | |
|---|--|
| <ul style="list-style-type: none">1) 게임<ul style="list-style-type: none">. 대전게임에서 상대해주는 에이전트. 게임 레벨 컨트롤2) 제어<ul style="list-style-type: none">. 차량의 거동 결정(차선을 바꾸고 싶은지, 핸들을 틀고 싶은지). 공장 로봇 자동화. 사람의 업무를 보조하는 로봇. 의수나 근육보조로봇. 서비스 로봇3) 자연어처리<ul style="list-style-type: none">. 챗봇의 좀 더 자연스러운 대화 | <ul style="list-style-type: none">4) 비전<ul style="list-style-type: none">. Object tracking. Segmentation 보조하는 에이전트5) 추천<ul style="list-style-type: none">. 실시간으로 사용자의 상황에 따라 추천(피드 추천, 광고 추천)6) 최적화<ul style="list-style-type: none">. 데이터센터 에너지 최적화. 최적 설계. Task scheduling. Neural Architecture Search |
|---|--|

* RLKorea 운영진의 의견

2. RL Tech. Tree

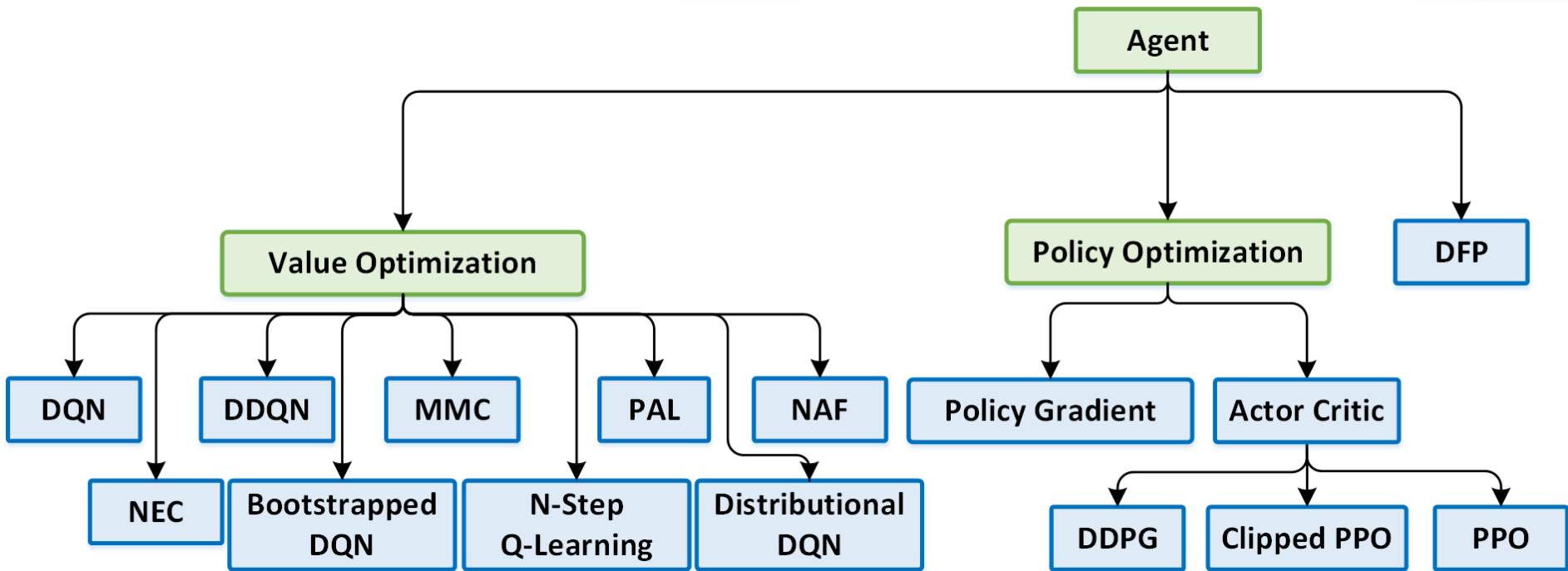


RL Algorithms Landscape



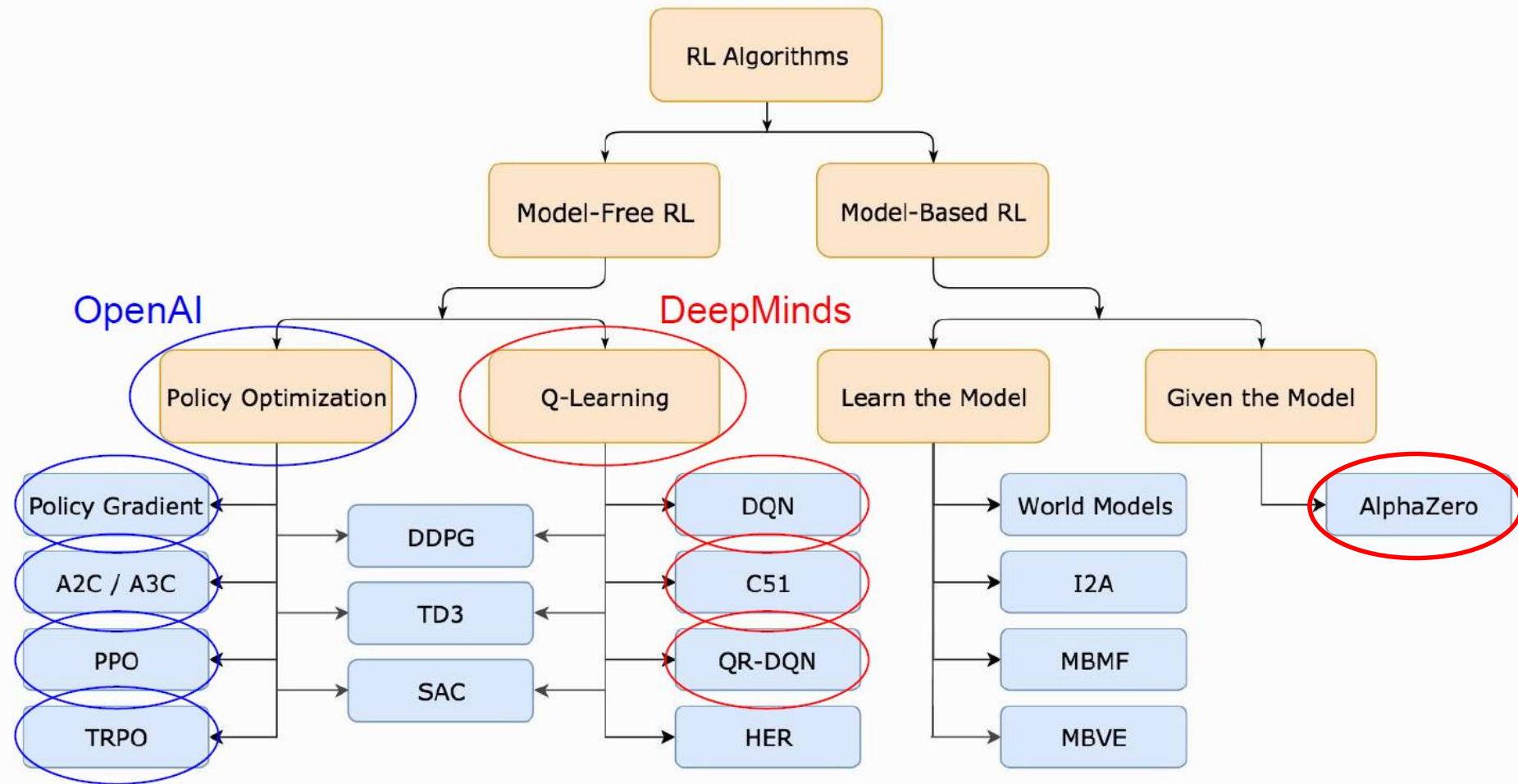
https://planspace.org/20170830-berkeley_deep_rl_bootcamp/

2. RL Tech. Tree



<https://stevenschmatz.gitbooks.io/deep-reinforcement-learning/content/>

2. RL Tech. Tree



* OpenAI의 주요 알고리즘 여행 및 적용 사례 소개, 플랜아이 차금강

3. Model-based RL vs. Model-free RL



What is the difference between model-based and model-free reinforcement learning?

To answer this question, let's revisit the components of an MDP, the most typical decision making framework for RL.

An MDP is typically defined by a 4-tuple (S, A, R, T) where

S is the state/observation space of an environment

A is the set of actions the agent can choose between

$R(s, a)$ is a function that returns the reward received for taking action a in state s

$T(s'|s, a)$ is a transition probability function, specifying the probability that the environment will transition to state s' if the agent takes action a in state s .

Our goal is to find a policy π that maximizes the expected future (discounted) reward.

<https://www.quora.com/What-is-the-difference-between-model-based-and-model-free-reinforcement-learning>

3. Model-based RL vs. Model-free RL

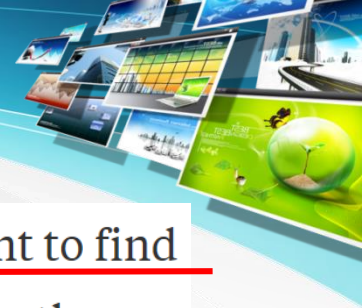


So, if the agent does not know the transition function T nor the reward function R , preventing it from planning a solution out, how can it find a good policy?

Well, it turns out there are lots of ways!

One approach that might immediately strike you, after framing the problem like this, is for the agent to learn a **model** of how the environment works from its observations and then plan a solution using that model. That is, if the agent is currently in state s_1 , takes action a_1 , and then observes the environment transition to state s_2 with reward r_2 , that information can be used to improve its estimate of $T(s_2|s_1, a_1)$ and $R(s_1, a_1)$, which can be performed using supervised learning approaches. Once the agent has adequately modelled the environment, it can use a planning algorithm with its learned model to find a policy. RL solutions that follow this framework are model-based RL algorithms.

3. Model-based RL vs. Model-free RL



As it turns out though, we don't have to learn a model of the environment to find a good policy. One of the most classic examples is *Q-learning*, which directly estimates the optimal Q -values of each action in each state (roughly, the utility of each action in each state), from which a policy may be derived by choosing the action with the highest Q -value in the current state. *Actor-critic* and *policy search* methods directly search over policy space to find policies that result in better reward from the environment. Because these approaches do not learn a model of the environment they are called *model-free algorithms*.

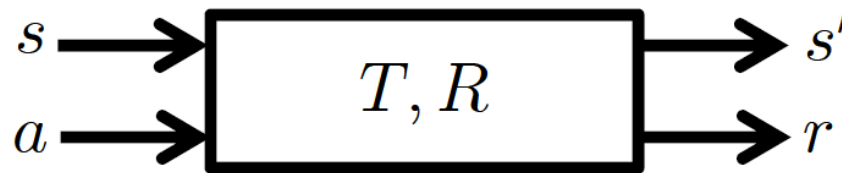
So if you want a way to check if an RL algorithm is model-based or model-free, ask yourself this question: after learning, can the agent make predictions about what the next state and reward will be before it takes each action? If it can, then it's a model-based RL algorithm. if it cannot, it's a model-free algorithm.

This same idea may also apply to decision-making processes other than MDPs.

<https://www.quora.com/What-is-the-difference-between-model-based-and-model-free-reinforcement-learning>

3. Model-based RL vs. Model-free RL

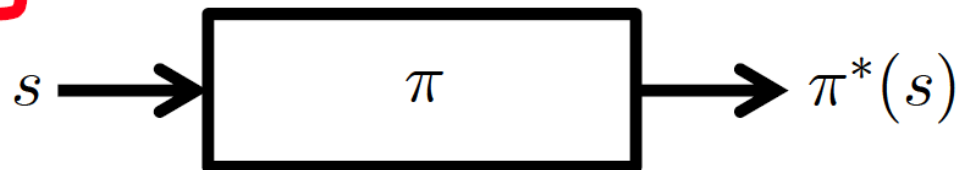
☐ Model-based algorithms



☐ Model-free (or Value-based) algorithms



☐ Policy search algorithms *



Littman, MLSS 2009

Model-based : MDP Learning

□ On experience $\langle s_t, a_t, r_t, s_{t+1} \rangle$:

- $R(s_t, a_t) \leftarrow R(s_t, a_t) + \alpha_t(r_t - R(s_t, a_t))$
- $T(s_t, a_t, s_{t+1}) \leftarrow T(s_t, a_t, s_{t+1}) + \alpha_t(1 - T(s_t, a_t, s_{t+1}))$
- $T(s_t, a_t, s') \leftarrow T(s_t, a_t, s') + \alpha_t(0 - T(s_t, a_t, s')) \quad \forall s' \neq s_{t+1}$

- $Q(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q(s', a')$ vs. SARSA

□ If:

- $\forall \langle s, a \rangle$ visited infinitely often
- $\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$

□ Then:

- $Q(s, a) \rightarrow Q^*(s, a)$ [Littman 1996]

가능한 모든 s' 를 탐색한 다음 것이 스트림스에 적합함.
[s, a] = [1 2 0 0 0, 9, 5] = [슬롯 + 새입력, 자리]
 $s' = [1 2 0 0 9 x]$ where $x \in \{3, 4, 5, 6, 7, 8, 10\}$
각 s' 마다 x 를 입력할 자리가 a' 가 됨.
예를 들어, $s' = [1 2 0 0 9, 3, 3]$ 에서 $x=4, a=3$.
이때 $Q(s', a')$ 를 모두 탐색하는 것이 중요.
T는 모든 s' 에 대해서 동일 즉, 어떤 다음 숫자가 나올지 모름.

Model-free (Value-based) : Q-Learning



□ On experience $\langle s_t, a_t, r_t, s_{t+1} \rangle$:

- $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t(r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t))$

시스템이 현재 결정한 유일한 상태
가 s' 와 해당하는 후보 액션
 a' 들에 대해서만 탐색함.

□ If:

- $\forall \langle s, a \rangle$ visited infinitely often
- $\sum_t \alpha_t = \infty, \sum_t \alpha_t^2 < \infty$

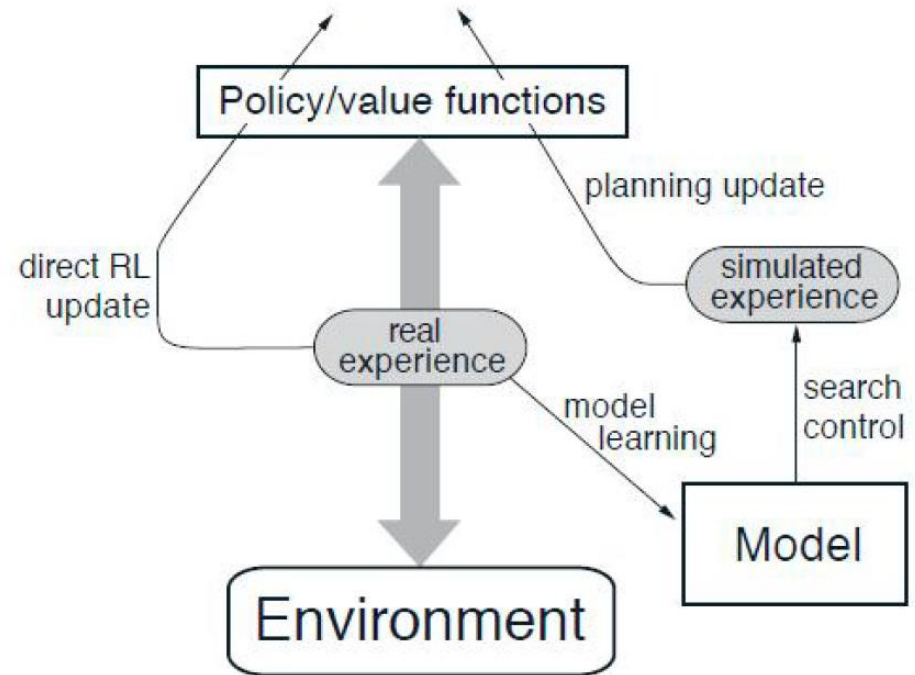
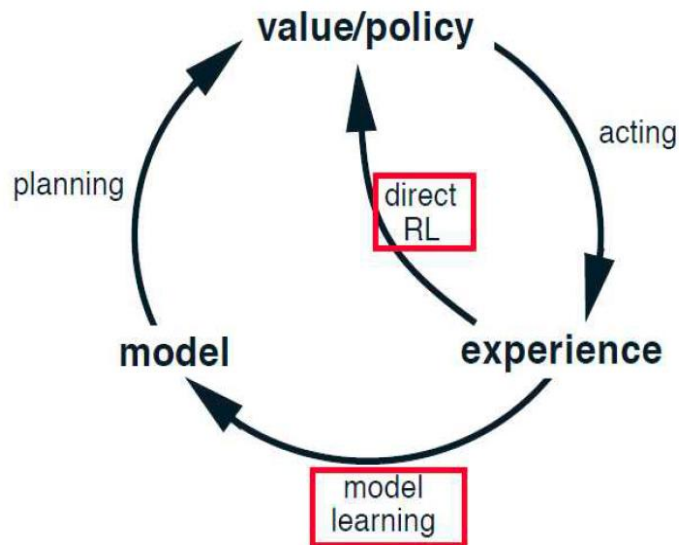
□ Then:

- $Q(s, a) \rightarrow Q^*(s, a)$ [Watkins & Dayan 1992]

Littman, MLSS 2009

Dyna : (Model-based) + (Model-free) = Integrated

Integrating Planning, Acting, and Learning



Model-Based Reinforcement Learning (MBRL)



- Model = simulator = dynamics = $T(s,a,s')$
 - may or may not include the reward function
- Model-free RL uses data from the environment only
- Model-based RL uses data from a model (which is given or estimated)
 - ◆ to use less data from the environment
 - ◆ to look ahead and plan
 - ◆ to explore
 - ◆ to guarantee safety
 - ◆ to generalize to different goals

Why MBRL now?



- Despite deep RL's recent success, it's still difficult to see its real-world applications
 - Requiring a huge amount of interactions (1M~1000M 🤖)
 - No safety guarantees
 - Difficult to transfer to other tasks
- MBRL can be a solution for these problems

MBRL Advantages vs. Disadvantages



Advantages:

- Can efficiently learn model by supervised learning methods
- Can reason about model uncertainty

Disadvantages:

- First learn a model, then construct a value function
⇒ two sources of approximation error

What is a Model?



- A *model* \mathcal{M} is a representation of an MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$, parametrized by η
- We will assume state space \mathcal{S} and action space \mathcal{A} are known
- So a model $\mathcal{M} = \langle \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$ represents state transitions $\mathcal{P}_\eta \approx \mathcal{P}$ and rewards $\mathcal{R}_\eta \approx \mathcal{R}$

$$\begin{aligned} S_{t+1} &\sim \mathcal{P}_\eta(S_{t+1} \mid S_t, A_t) \\ R_{t+1} &= \mathcal{R}_\eta(R_{t+1} \mid S_t, A_t) \end{aligned}$$

- Typically assume **conditional independence** between state transitions and rewards

$$\mathbb{P}[S_{t+1}, R_{t+1} \mid S_t, A_t] = \mathbb{P}[S_{t+1} \mid S_t, A_t] \mathbb{P}[R_{t+1} \mid S_t, A_t]$$

Model Learning



- Goal: estimate model \mathcal{M}_η from experience $\{S_1, A_1, R_2, \dots, S_T\}$
- This is a supervised learning problem

$$S_1, A_1 \rightarrow R_2, S_2$$

$$S_2, A_2 \rightarrow R_3, S_3$$

$$\vdots$$

$$S_{T-1}, A_{T-1} \rightarrow R_T, S_T$$

- Learning $s, a \rightarrow r$ is a regression problem
- Learning $s, a \rightarrow s'$ is a density estimation problem
- Pick loss function, e.g. mean-squared error, KL divergence, ...
- Find parameters η that minimise empirical loss

Examples of Models



- Table Lookup Model
- Linear Expectation Model
- Linear Gaussian Model
- Gaussian Process Model
- Deep Belief Network Model
- ...

Table Lookup Model



- Model is an explicit MDP, $\hat{\mathcal{P}}, \hat{\mathcal{R}}$
- Count visits $N(s, a)$ to each state action pair

$$\hat{\mathcal{P}}_{s,s'}^a = \frac{1}{N(s, a)} \sum_{t=1}^T \mathbf{1}(S_t, A_t, S_{t+1} = s, a, s')$$
$$\hat{\mathcal{R}}_s^a = \frac{1}{N(s, a)} \sum_{t=1}^T \mathbf{1}(S_t, A_t = s, a) R_t$$

- Alternatively
 - At each time-step t , record experience tuple $\langle S_t, A_t, R_{t+1}, S_{t+1} \rangle$
 - To sample model, randomly pick tuple matching $\langle s, a, \cdot, \cdot \rangle$

Planning with a Model



- Given a model $\mathcal{M}_\eta = \langle \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$
- Solve the MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$
- Using favourite planning algorithm
 - Value iteration
 - Policy iteration
 - Tree search
 - ...

Planning with an Inaccurate Model



- Given an imperfect model $\langle \mathcal{P}_\eta, \mathcal{R}_\eta \rangle \neq \langle \mathcal{P}, \mathcal{R} \rangle$
- Performance of model-based RL is limited to optimal policy for approximate MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}_\eta, \mathcal{R}_\eta \rangle$
- i.e. Model-based RL is only as good as the estimated model
- When the model is inaccurate, planning process will compute a suboptimal policy
- Solution 1: when model is wrong, use model-free RL
- Solution 2: reason explicitly about model uncertainty

4. 몇가지 사례들 (1)



ICRA 2018 Spotlight Video

NEURAL NETWORK DYNAMICS FOR MODEL-BASED DEEP REINFORCEMENT LEARNING WITH MODEL-FREE FINE-TUNING

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, Sergey Levine
UC Berkeley

<https://www.youtube.com/watch?v=G7IXiuEC8x0&feature=share>



MODEL-BASED (OURS)

REWARD: 1000

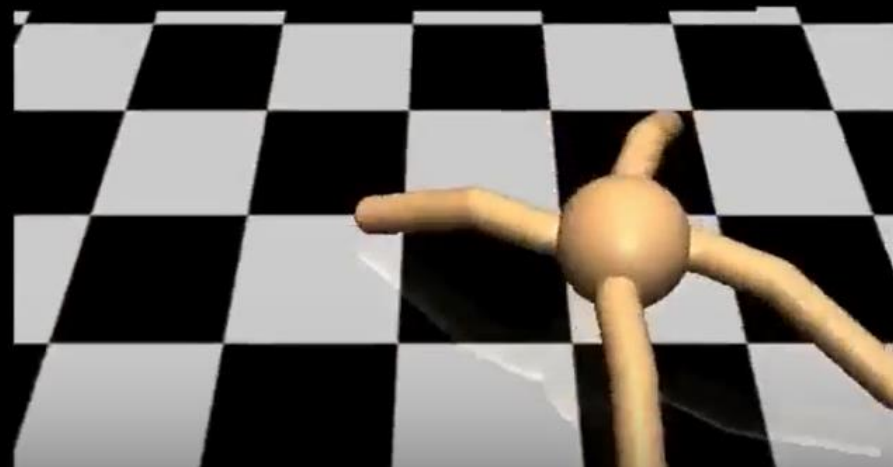
* DATAPOINTS USED:
400,000



MODEL-FREE

* REWARD: 5000

DATAPOINTS USED:
25,000,000





RUN OUR
MODEL-BASED
APPROACH

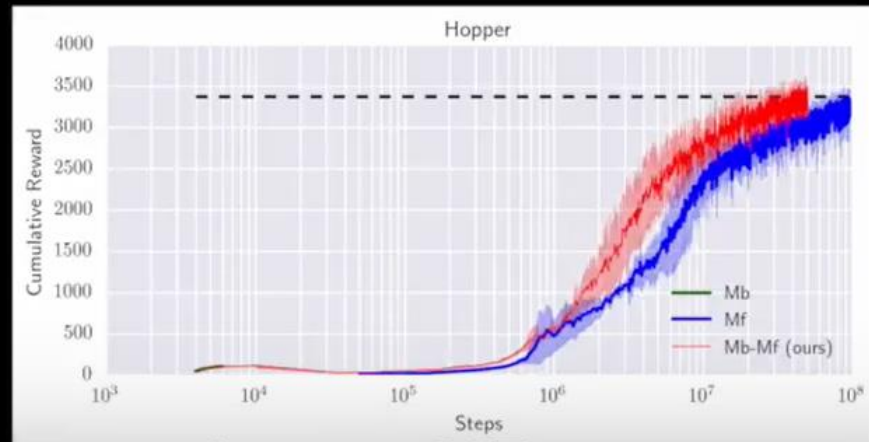
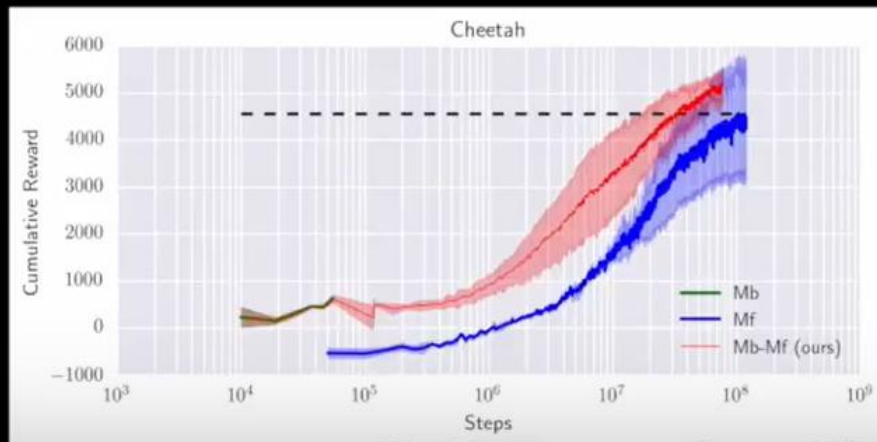
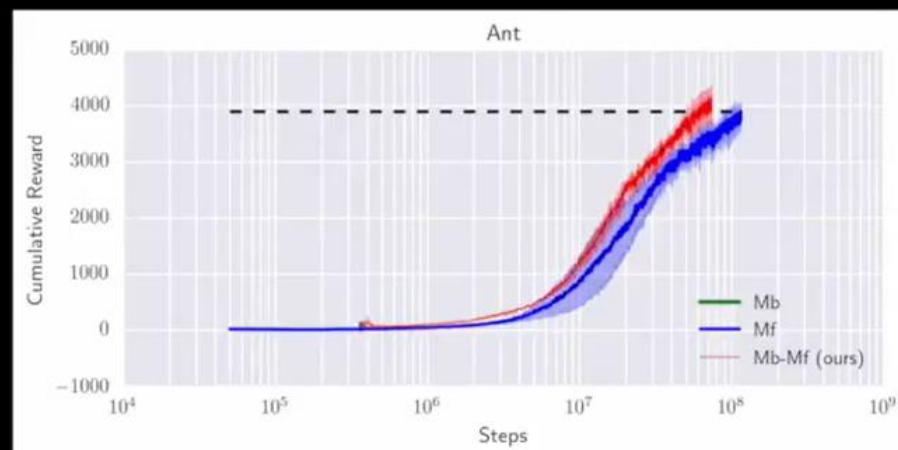
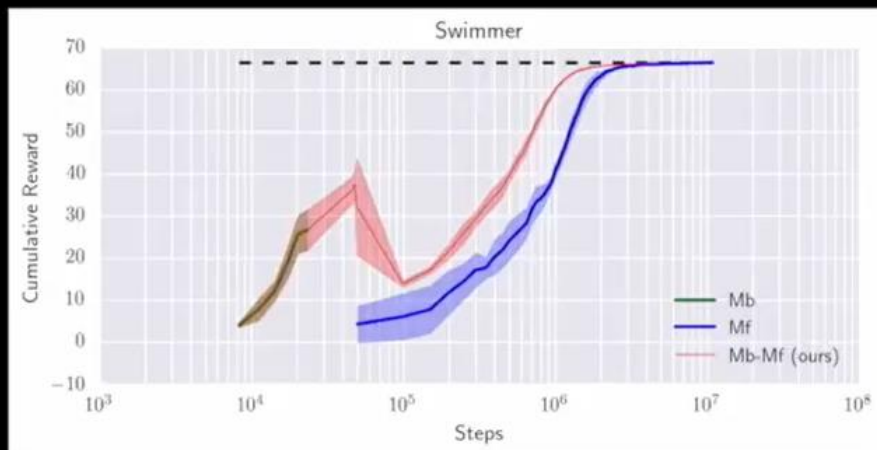


TRAIN POLICY π_θ
TO IMITATE
MODEL-BASED
CONTROLLER



RUN
MODEL-FREE
ALGORITHM ON
PRE-INITIALIZED
POLICY π_θ





3-5x sample efficiency gain over MF

4. 몇가지 사례들 (2)

ICLR 2018

Temporal Difference Models: Deep Model-free RL for Model-based RL

Shixiang (Shane) Gu* (顾世翔)

(work done at Google internship)

Co-authors: Victhyr Pong*, Murtaza Dalal, Sergey Levine (*joint first-author)



<https://www.youtube.com/watch?v=j-3nUkzMFA8&feature=share>

Model-free Deep RL: limitations

“Model-free”: High sample intensity



“Deep” may not be needed (for gym benchmarks)

- linear policies [Rajeswaran et. al., 2017]
- maybe nearest neighbor [Mansimov et. al., 2017]



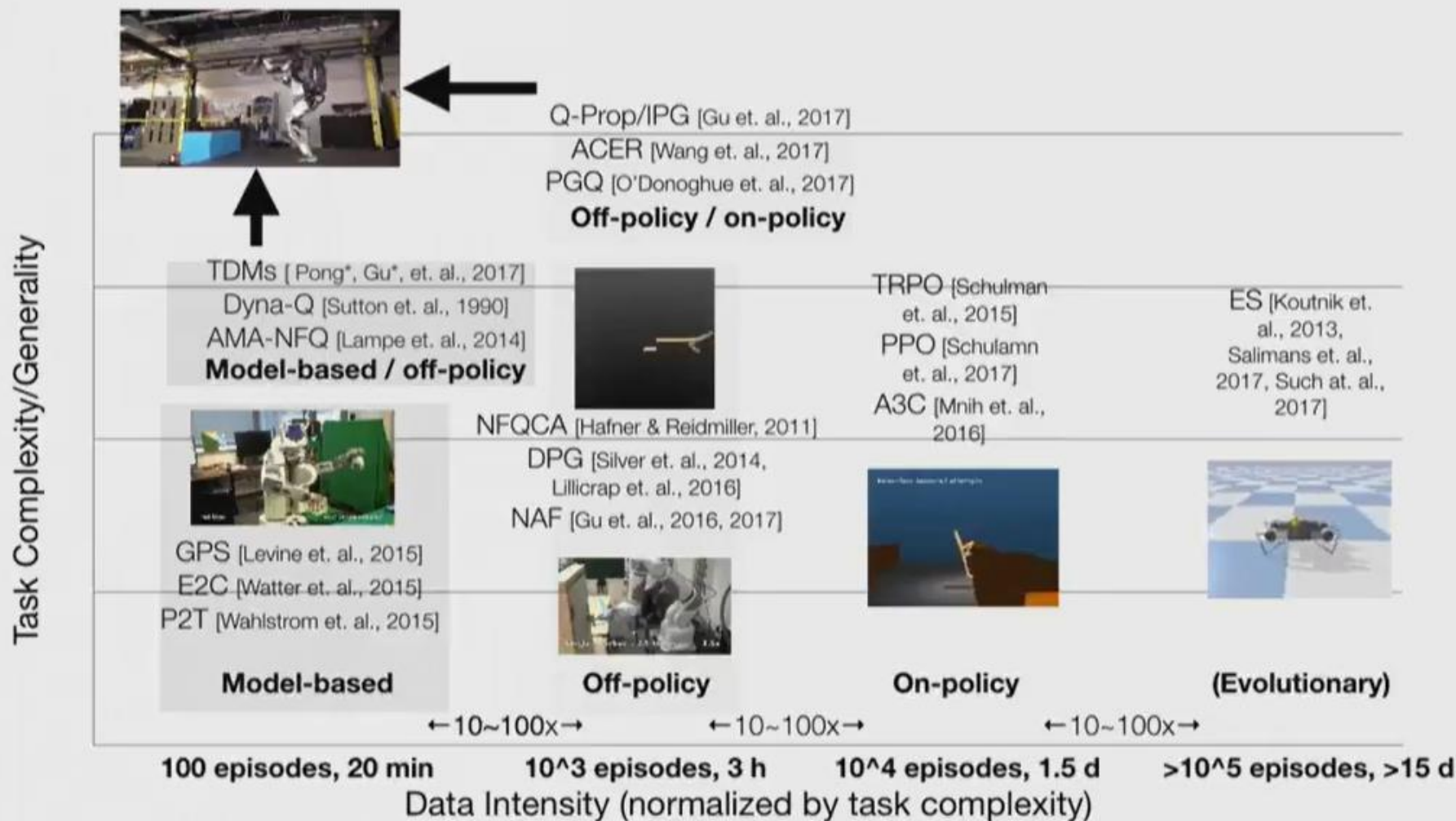
“RL” may not be needed

- Evolutionary Strategies [Koutnik et. al., 2013, Salimans et. al., 2017, Such et. al., 2017]

Potential cause: model-free RL lacks rich learning signals



Sample-efficiency of Deep RL methods





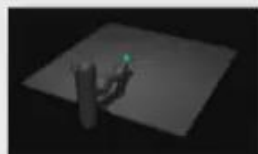
Problem statements

How can we add more learning signals to model-free RL?

- Prior work:
 - Auxiliary tasks (UNREAL) [Jaderberg et. al., 2017]
 - Act by prediction [Dosovitskiy et. al., 2017]
 - Intrinsic rewards [Mohamed et. al., 2015; etc.]
 - **Universal value functions/Hindsight experience replay (HER)** [Schaul et. al., 2015; Andrychowicz et. al., 2017]

Can model-free RL leverage **as much or more** learning signals than model-based?

Experiments: Benchmarking



(a) 7-DoF Reacher



(b) Pusher



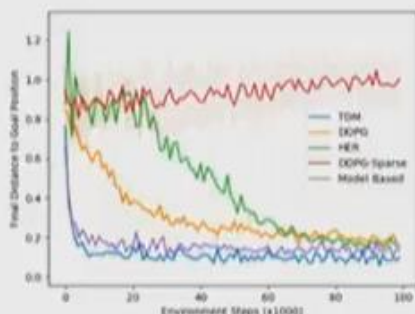
(c) Half Cheetah



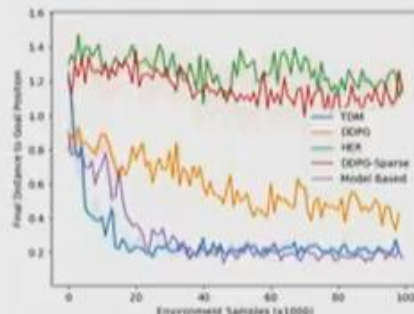
(d) Ant



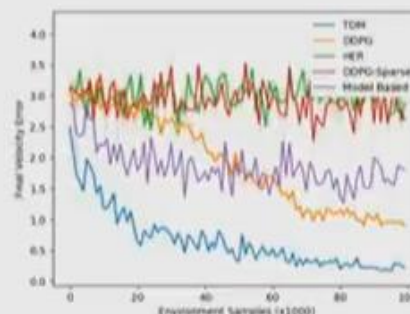
(e) Sawyer Robot



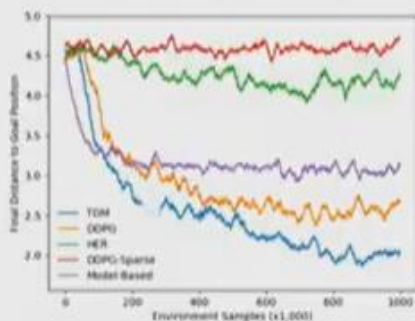
(a) 7-Dof Reacher



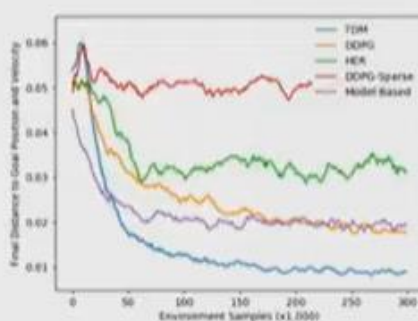
(b) Pusher



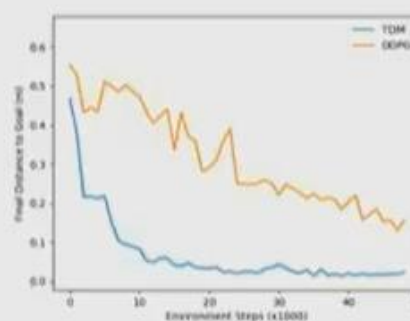
(c) Half Cheetah



(d) Ant: Position



(e) Ant: Position and Velocity



(f) Sawyer Robot (Real-world)

- TDMs are significantly more sample efficient than classic model-free methods
- TDMs do not incur performance drop as classic model-based methods

Discussion

Infinite cherries >>> Cake

Important direction: **redefine model-free RL**,
e.g. **how to increase supervision signals**

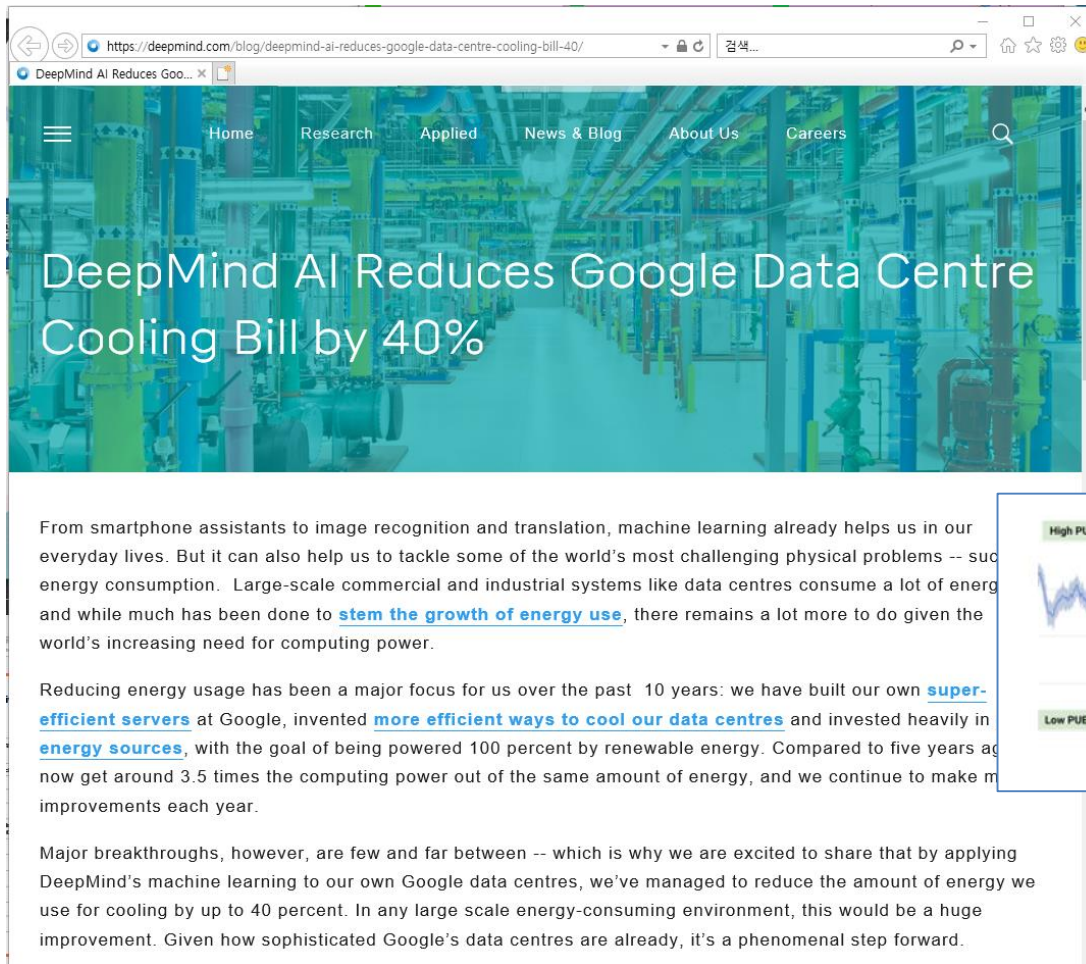
- Per task: UVF [Schaul et. al., 2015], HER [Andrychowicz et. al., 2017], TDMS, Distributional RL [Bellemare et. al., 2017]
- Through many tasks: Meta learning & transfer learning (from simulation)
- **Stable RL optimization objective with lots of learning signals = Success is guaranteed**



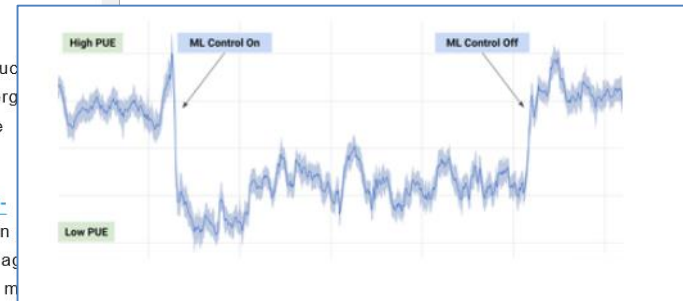
RL problems are exciting: many problems and future potentials

4. 몇가지 사례들 (3)

Google DeepMind



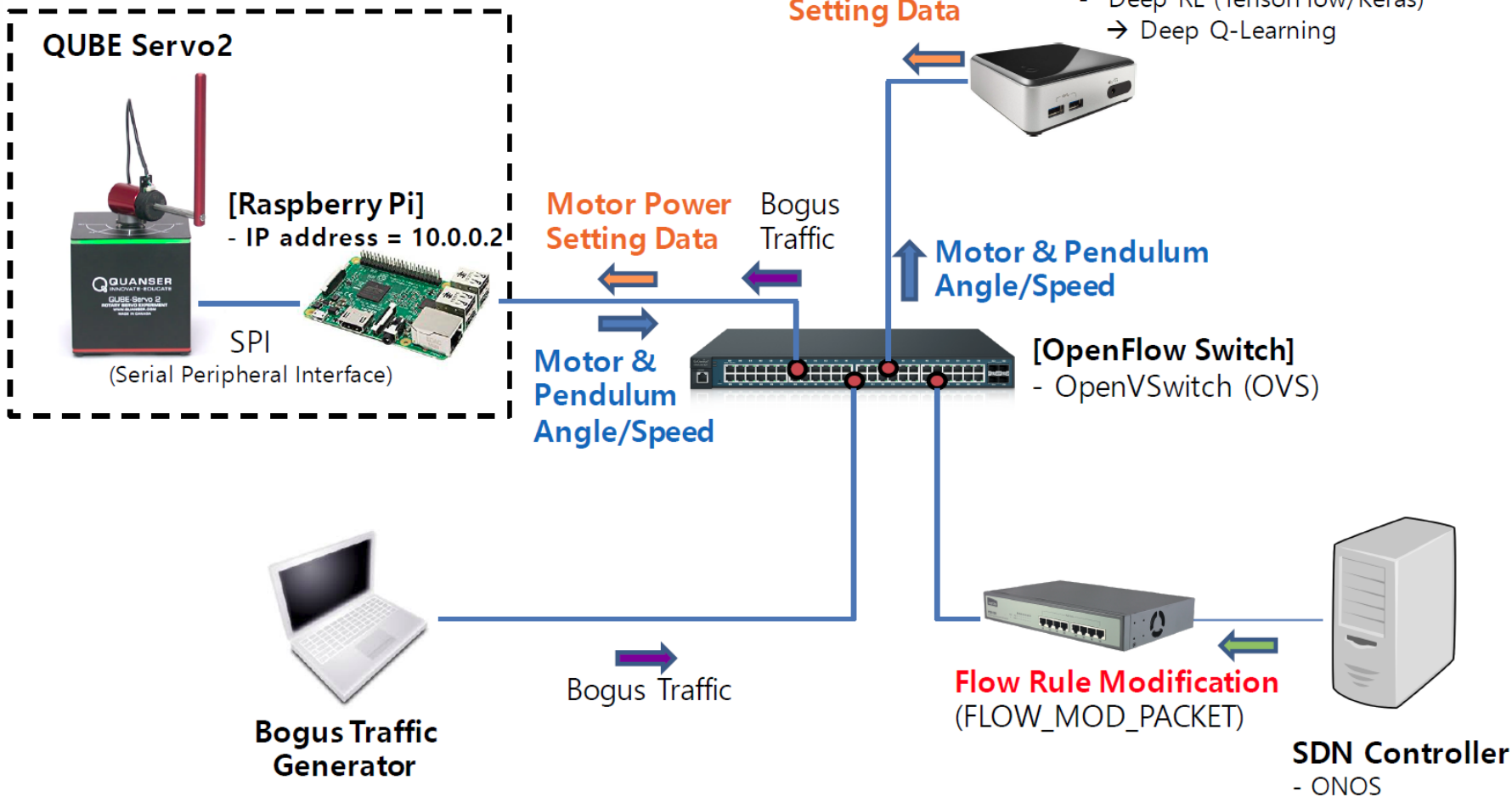
The screenshot shows a web browser window displaying the DeepMind blog post. The URL is <https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/>. The page features a navigation bar with links to Home, Research, Applied, News & Blog, About Us, and Careers. The main heading is "DeepMind AI Reduces Google Data Centre Cooling Bill by 40%". Below the heading, the text reads: "From smartphone assistants to image recognition and translation, machine learning already helps us in our everyday lives. But it can also help us to tackle some of the world's most challenging physical problems -- such as energy consumption. Large-scale commercial and industrial systems like data centres consume a lot of energy and while much has been done to [stem the growth of energy use](#), there remains a lot more to do given the world's increasing need for computing power." The next paragraph states: "Reducing energy usage has been a major focus for us over the past 10 years: we have built our own [super-efficient servers](#) at Google, invented [more efficient ways to cool our data centres](#) and invested heavily in [energy sources](#), with the goal of being powered 100 percent by renewable energy. Compared to five years ago, we now get around 3.5 times the computing power out of the same amount of energy, and we continue to make improvements each year." The final paragraph mentions: "Major breakthroughs, however, are few and far between -- which is why we are excited to share that by applying DeepMind's machine learning to our own Google data centres, we've managed to reduce the amount of energy we use for cooling by up to 40 percent. In any large scale energy-consuming environment, this would be a huge improvement. Given how sophisticated Google's data centres are already, it's a phenomenal step forward."



<https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/>

4. 몇가지 사례들 (4-1)

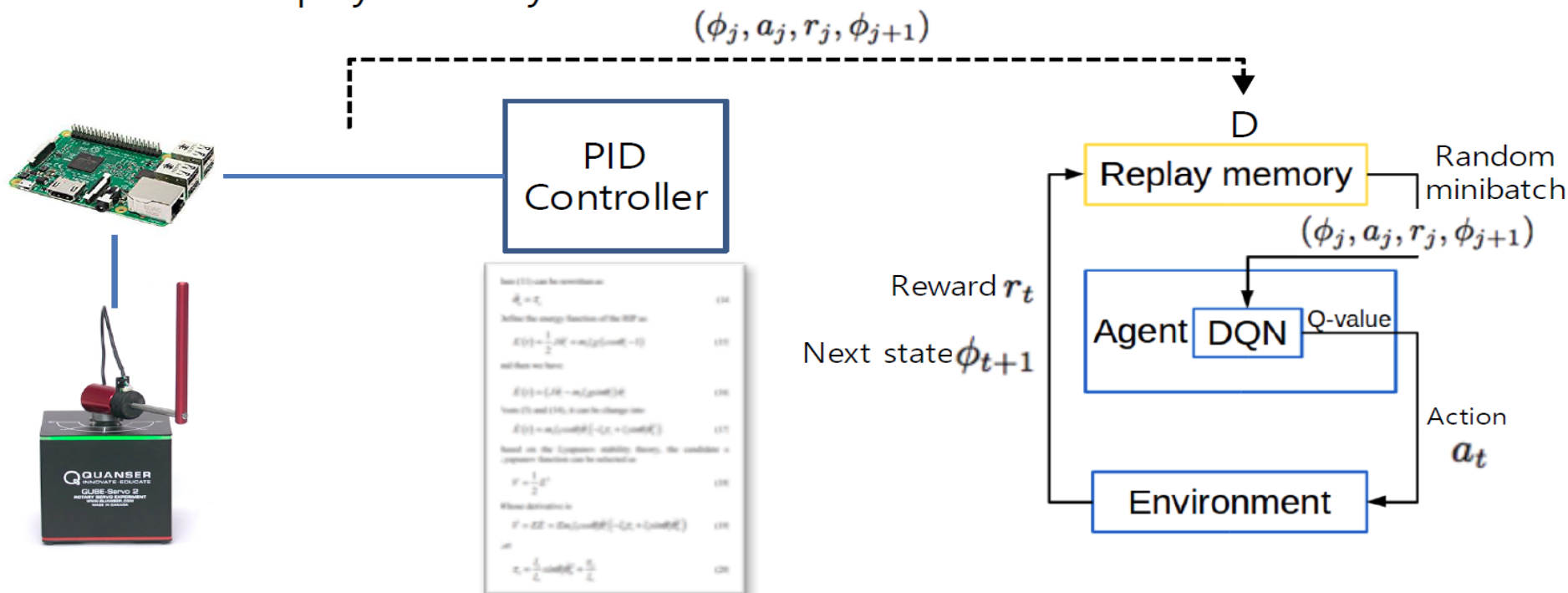
ETRI KSB융합연구단+한기대



Imitational RL (모방 강화 학습)

◆ Learning Acceleration

- with help of classical PID control model
- Fill up a large number (20,000) of good transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ into the replay memory



5. Summary



- MBRL is hot
 - There were more papers than I can introduce
- Popular ideas
 - Incorporating a model/planning structure into a NN
 - Use model-based simulations to reduce sample complexity
- (Deep) MBRL can be a solution to drawbacks of deep RL
- However, MBRL has its own challenges
 - How to learn a good model
 - How to make use of a possibly bad model

발표를 마치면서,,

대덕 연구단지를 함께 AI 메카로~
대한민국을 함께 AI 메카로~

