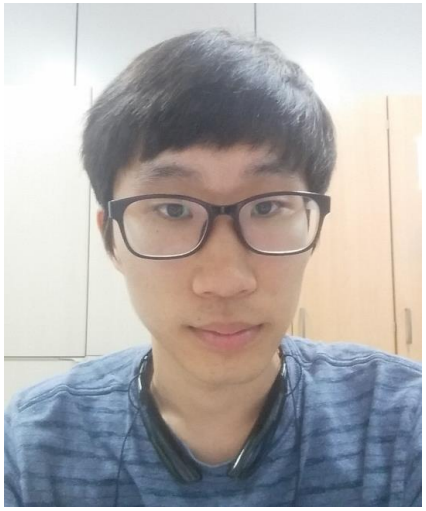
강화학습

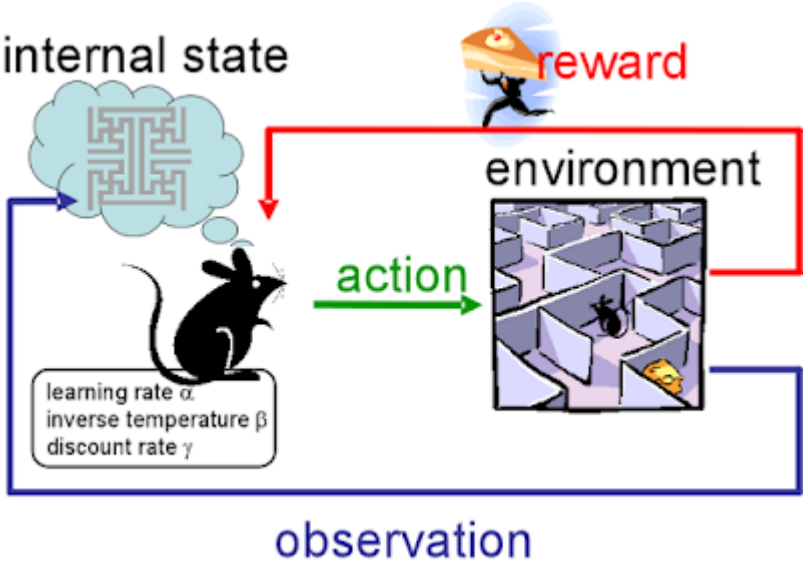
Idea Suggestions

KeumGang Cha
https://gitbub.com/chagmgang

소개



- 부산대학교 기계공학부 학사
- 부산대학교 기계공학부 제어자동화시스템공학과 석사
- 한국생산기술연구원 연구원
- 한국과학기술원 박사과정 (휴학)
- ㈜플랜아이 미래연구소 연구원
- 강화학습을 취미로 하는 사람

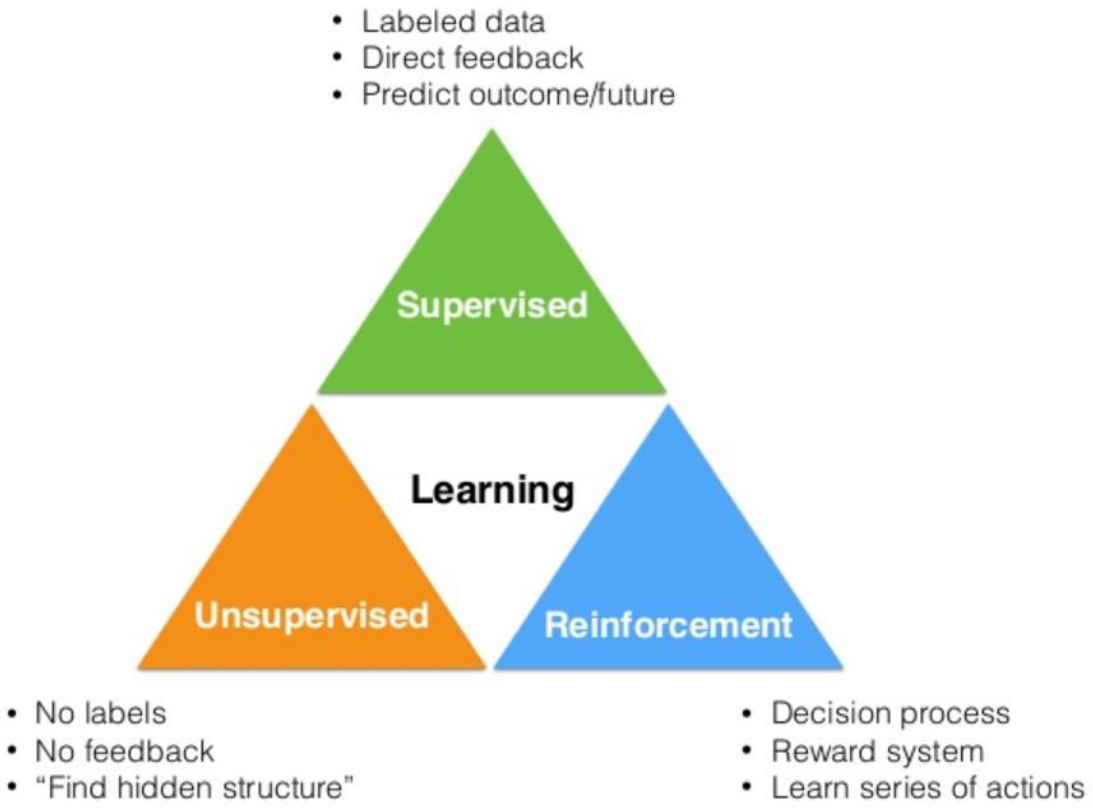


딥러닝

Idea Suggestions

KeumGang Cha

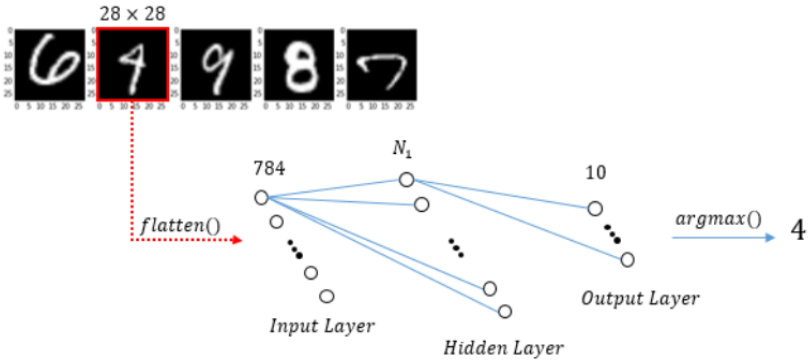
딥러닝


• Labeled data
• Direct feedback
• Predict outcome/future
Supervised
Learning
Unsupervised
Reinforcement
• No labels
• No feedback
• “Find hidden structure”
• Decision process
• Reward system
• Learn series of actions

딥러닝

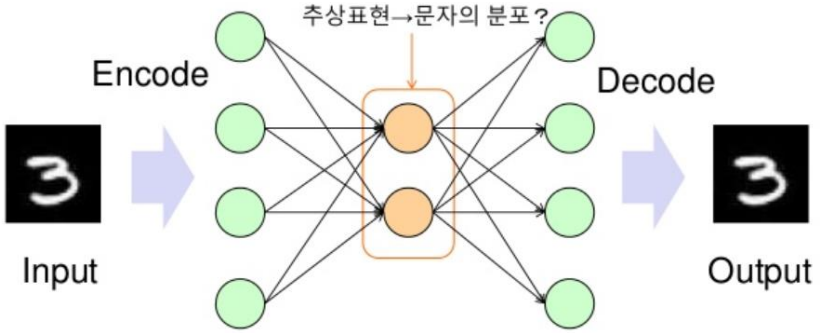
❑ 지도 학습

- 무엇인지 알고 싶을 때
- 어디 있는지 알고 싶을 때 등등..
- 사람으로 따지면 시각, 촉각, 후각, 미각



❑ 비지도 학습

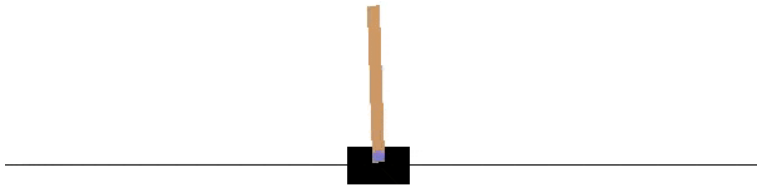
- 생성하고 싶을 때
- 답이 없는 것을 나누고 싶을 때
- 상상하고 싶을 때
- 비슷한 것을 만들어내고 싶을 때
- 사람으로 따지면 상상, 추론..



딥러닝

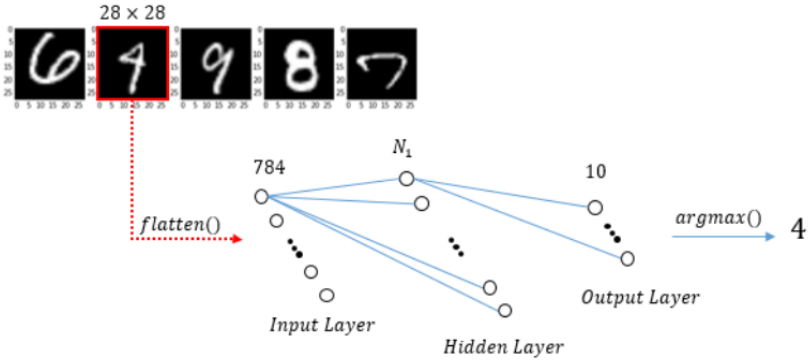
❑ 강화학습

- 어떻게 행동해야 되는지 알고 싶을 때
- 이 행동으로 인해 다음 상태가 어떻게 될지 알고 싶을 때
- 원하는 업무를 수행하도록 하고 싶을 때

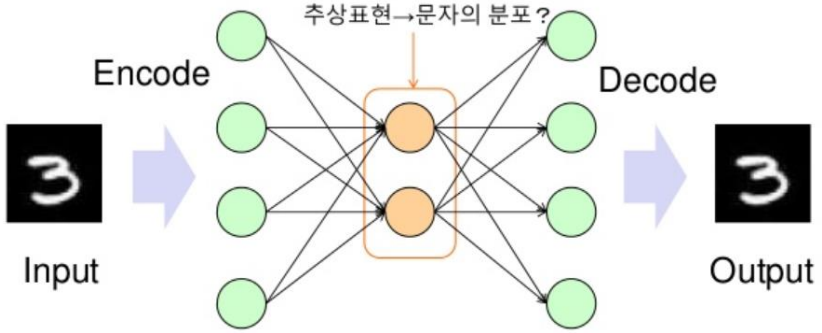


강화학습의 차이점

❑ 지도 학습

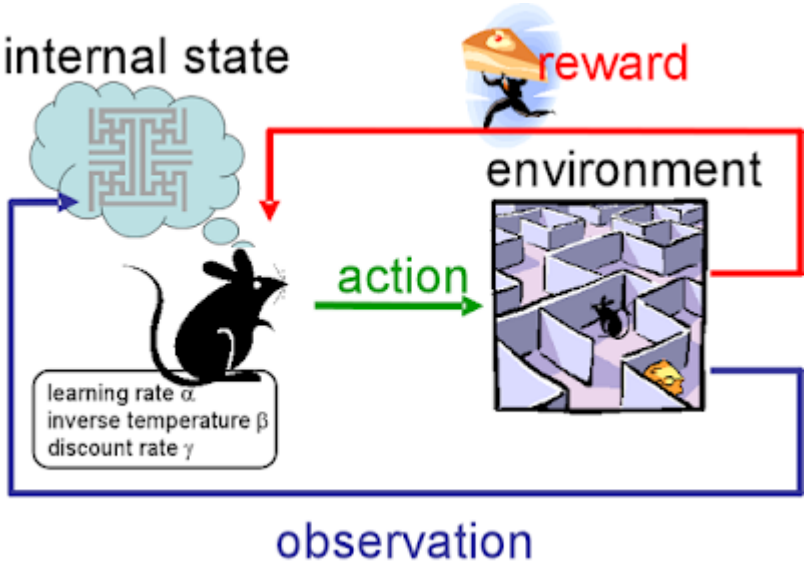


❑ 비지도 학습



❑ 강화학습

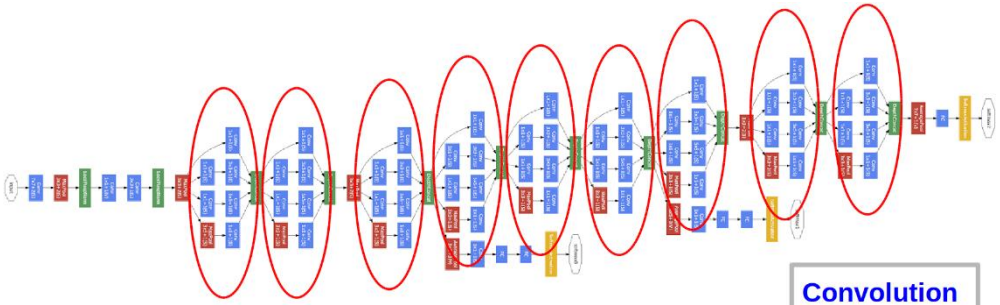
- 환경과의 상호작용
- 답을 스스로 도출
- 입력 값에 대한 보상만 주어짐



지도 학습의 현주소

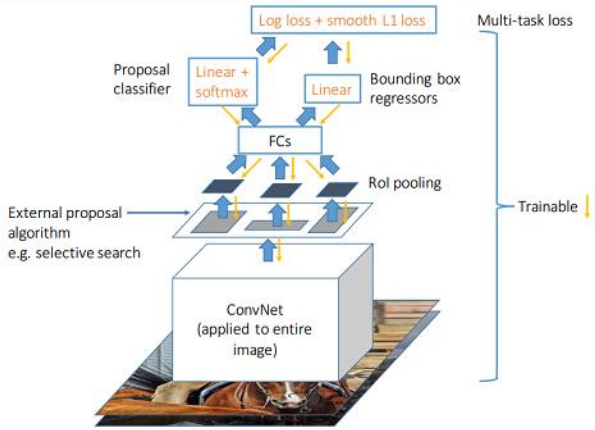
❑ 이미지 분류 및 위치 검출

- 분류 : LeNet, AlexNet(Inception), VGG, GoogLeNet, ResNet...
- 분할 : RPN(Region Proposal Network)...
- 분류 + 분할 : RCNN, Fast-RCNN, Faster-RCNN

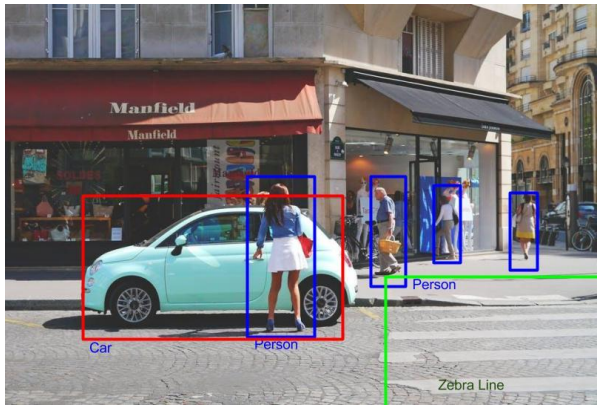


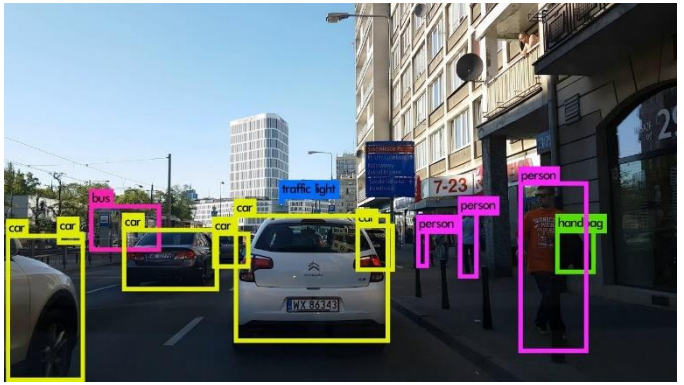
GoogLeNet Architecture

Convolution
Pooling
Softmax
Concat/Normalize



Faster-RCNN





비지도 학습의 현주소

❑ 데이터의 재생성 및 스타일 분류

- starGAN, waveGAN, cycleGAN, waveGAN ....
- AutoEncoder, Denoising Network, Unet...

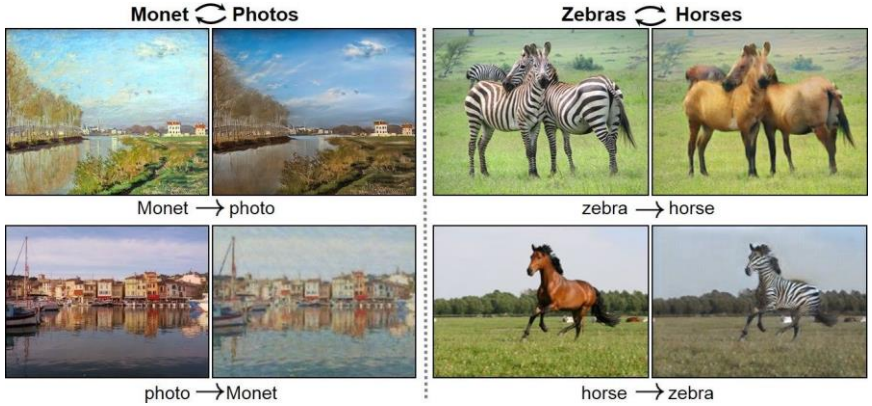


Image Style Transfer

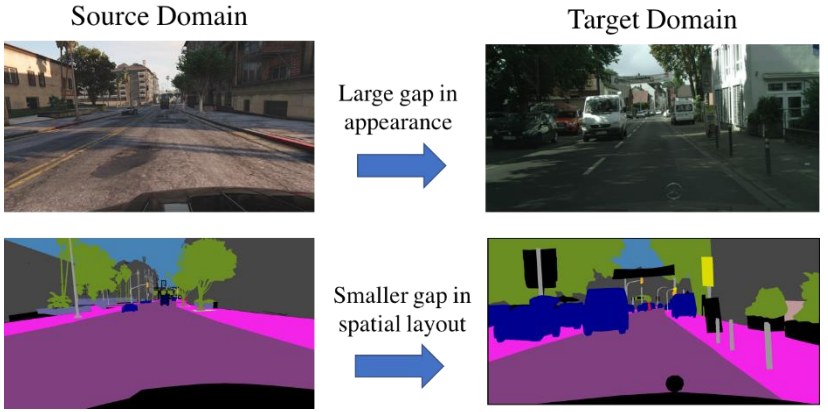
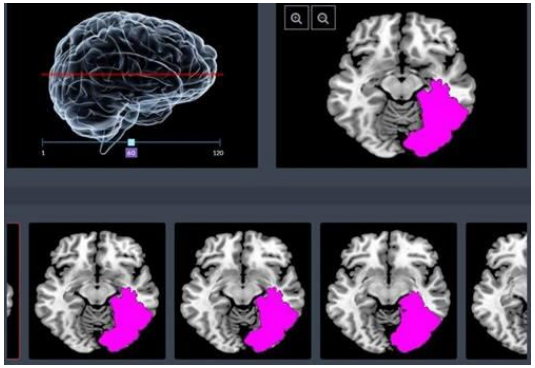


Image Segmentation

waveGAN

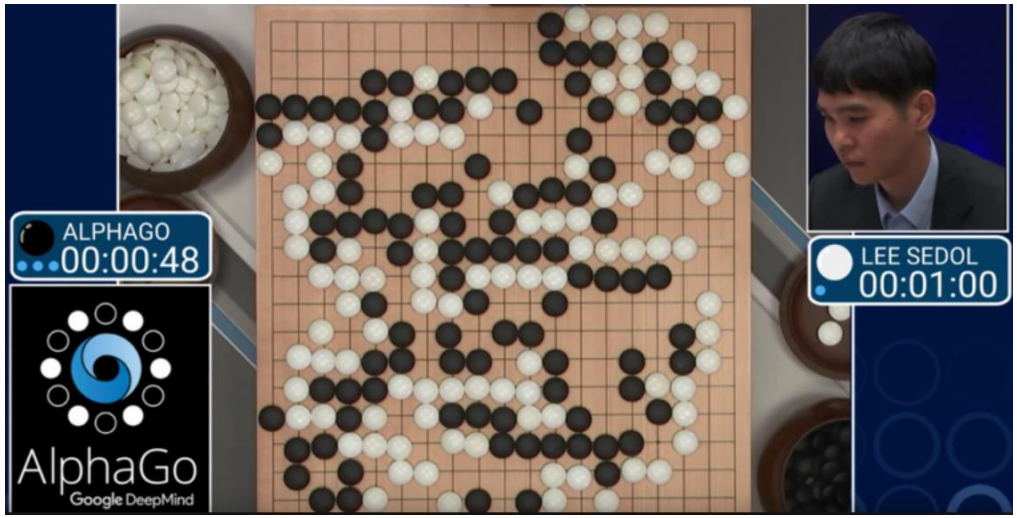


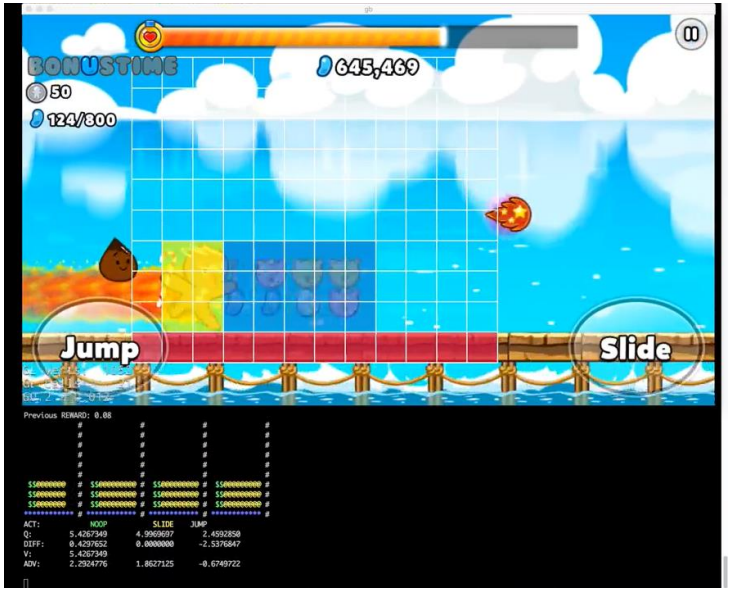
Medical Analysis

강화학습의 현주소

❑ 원하는 대상의 최적화

- Q-Network(Double, Dueling, DD)
- Prioritized Experience Replay
- Policy Gradient, ...



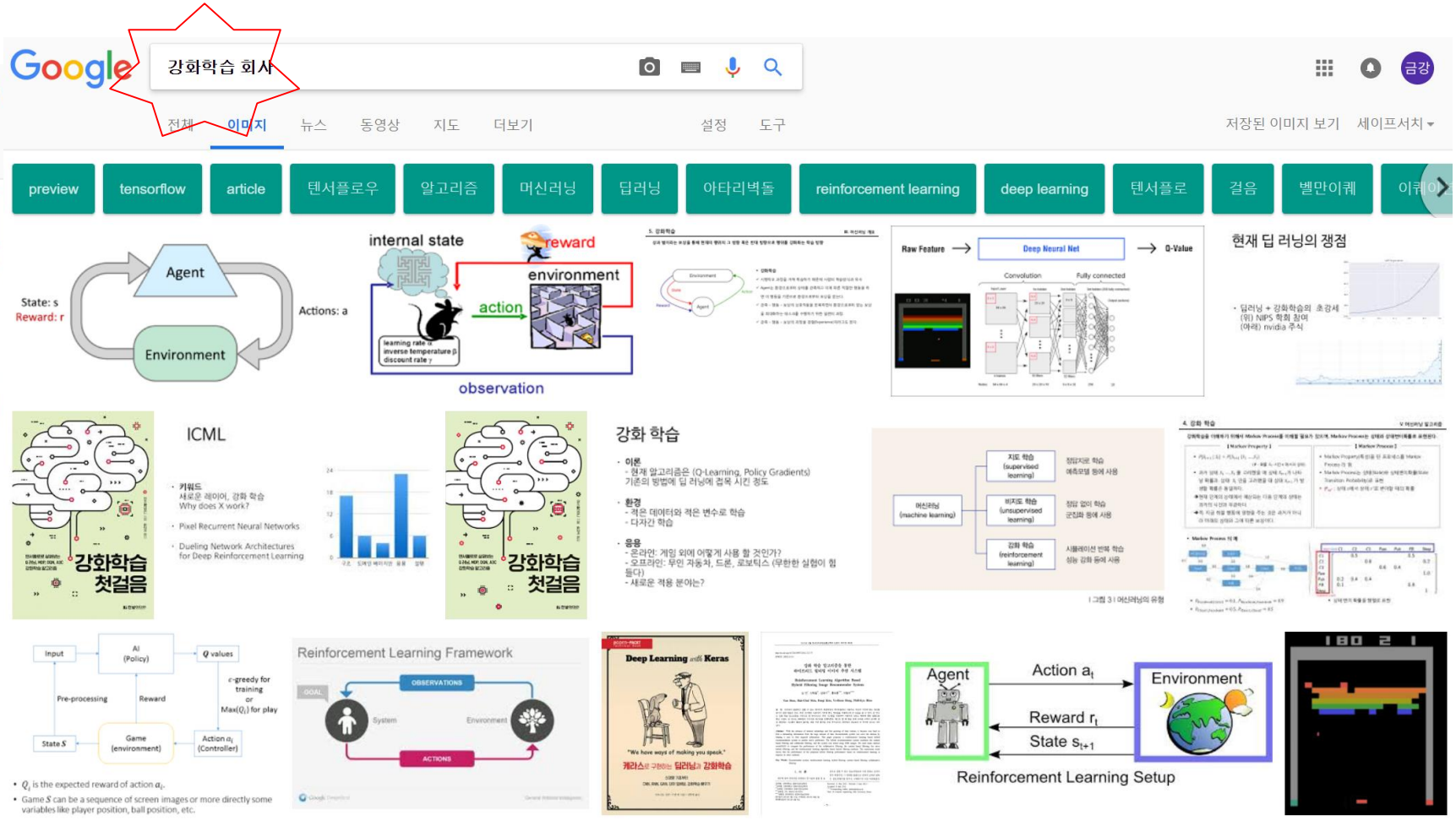


왜 강화학습을 해야하는가

Idea Suggestions

KeumGang Cha

희소성


Google
강화학습 회사
전체
이미지
뉴스
동영상
지도
더보기
설정
도구
저장된 이미지 보기
세이프서치
금강
preview
tensorflow
article
텐서플로우
알고리즘
머신러닝
딥러닝
아타리벽돌
reinforcement learning
deep learning
텐서플로
걸음
벨만이퀘
Agent
Environment
State: s
Reward: r
Actions: a
internal state
reward
environment
action
observation
Raw Feature
Deep Neural Net
Q-Value
Convolution
Fully connected
현재 딥 러닝의 쟁점
강화학습 첫걸음
ICML
강화학습 첫걸음
강화 학습
머신러닝 (machine learning)
지도 학습 (supervised learning)
비지도 학습 (unsupervised learning)
강화 학습 (reinforcement learning)
Input
AI (Policy)
Q values
Pre-processing
Reward
Game (environment)
Action a_t (Controller)
Reinforcement Learning Framework
OBSERVATIONS
ACTIONS
System
Environment
Deep Learning with Keras
Agent
Action a_t
Environment
Reward r_t
State s_{t+1}
Reinforcement Learning Setup

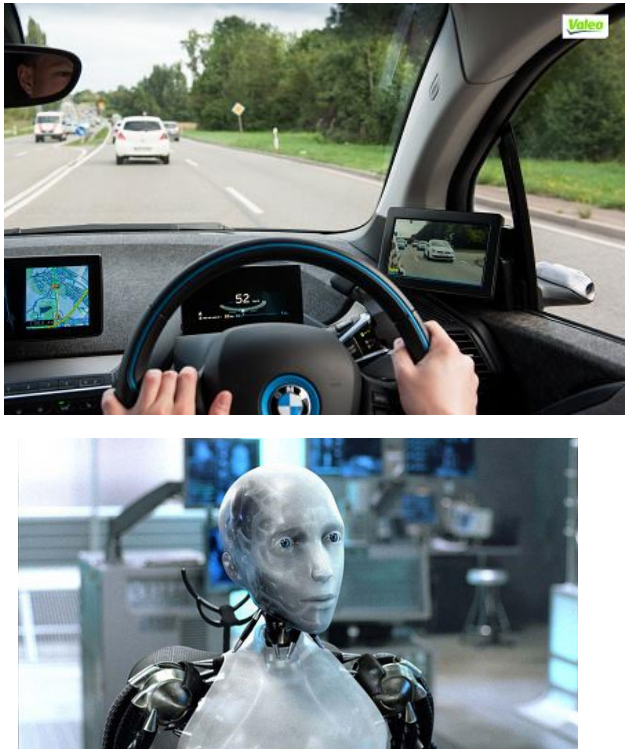
인공지능의 단계

❑ 인지 & 판단

- **주변의 상황을 인지**
  - **차선, 신호등, 보행자, 속도, 가속도...**
- **상황에 따른 판단**
  - **휠, 제동, 가속...**

지도학습

강화학습

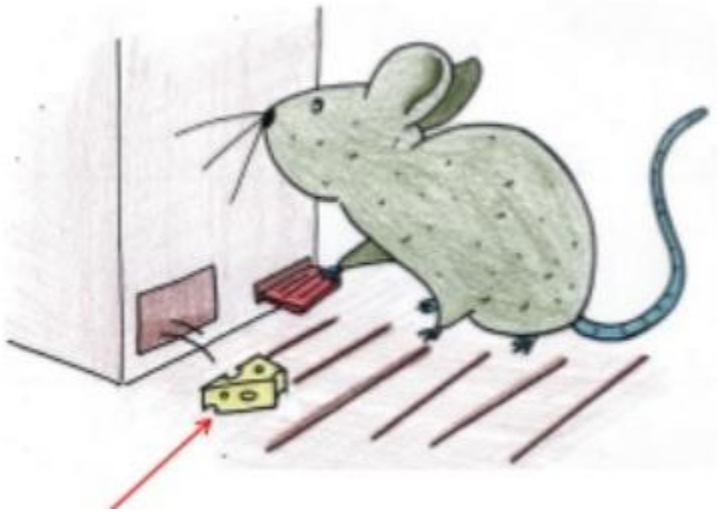


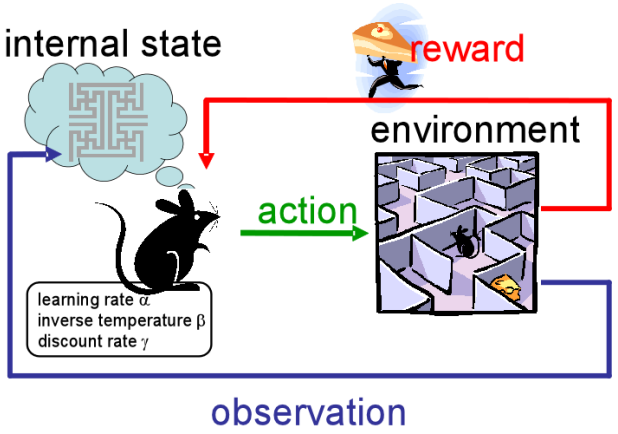
강화학습이란?

Idea Suggestions

KeumGang Cha

강화학습은?





출처 : 파이썬과 케라스로 배우는 강화학습(저자:이웅원)

환경을 정의해보자

❑ Markov Decision Process

Markov Decision Process is Markov reward process with decisions. It is environment in which all states are Markov.

- **A Markov Decision Process is a set of** $< S, P, A, R, \gamma >$
- P **is a probability set of**
- S **is a finite set of states**
- A **is a finite set of actions**
- R **is reward function,** $R_s^a = E[R_{t+1}|S_t = s, A_t = a]$
- γ **is a discount factor** $\gamma \in [0,1]$

❑ Bellman Equation

- $Q(s_t, a_t) = R_s^a + \gamma \max Q(s_{t+1}, a_{t+1})$

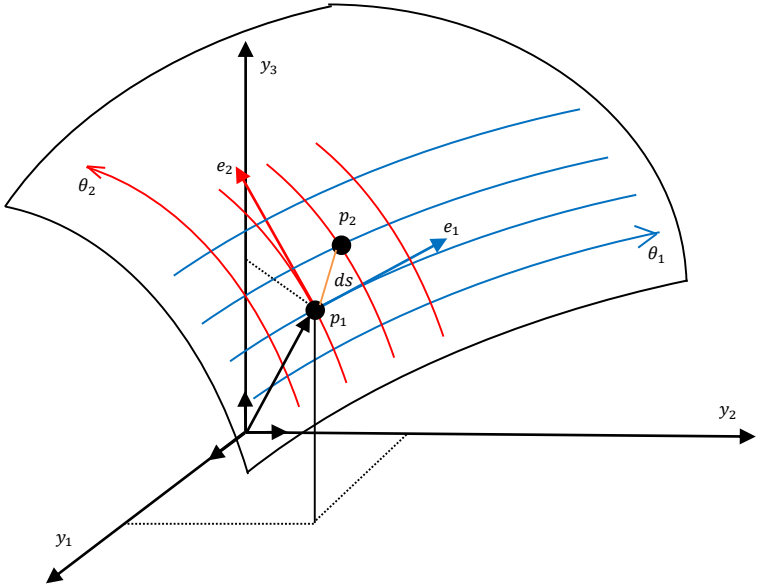
알고리즘

❑ 가치 최적화

- $action = \underset{a}{argmax}\, Q(s_t, a_t)$
- **현재 상태에서 가장 가치가 높은 행동을 선택**

❑ 정책 최적화

- $maximize\, J(\theta) = \mathrm{E}[\log \pi_\theta(a|s)\, Q(s_t, a_t)]$
- $\pi_\theta(a|s)$ **: 현재 상태에서 어떠한 행동을 선택할 확률**
- $J(\theta)$ **: 기대 보상 값**



길찾기에 적용해보자

❑ 가치최적화(다이나믹 프로그래밍)

- γ **is a discount factor** $\gamma \in [0,1]$
- $Q(s_t, a_t) = R_s^a + \gamma Q(s_{t+1}, a_{t+1})$

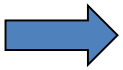
	1	2	3	…
Up	0	0	0	
Down	0	0	0	
Left	0	0	0	
Right	0	0	0	

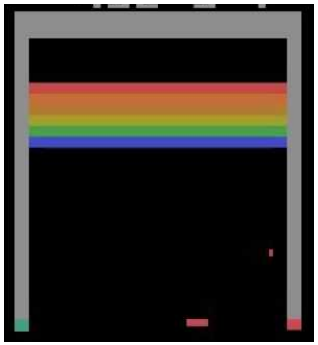
❑ 행렬로 풀 수 있는 문제

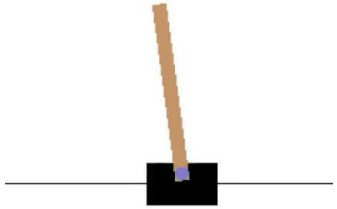
❑ 딥러닝과 무슨 관계가 있는 것인가

없습니다.

❑ 강화학습으로 다른 문제를 풀어보자







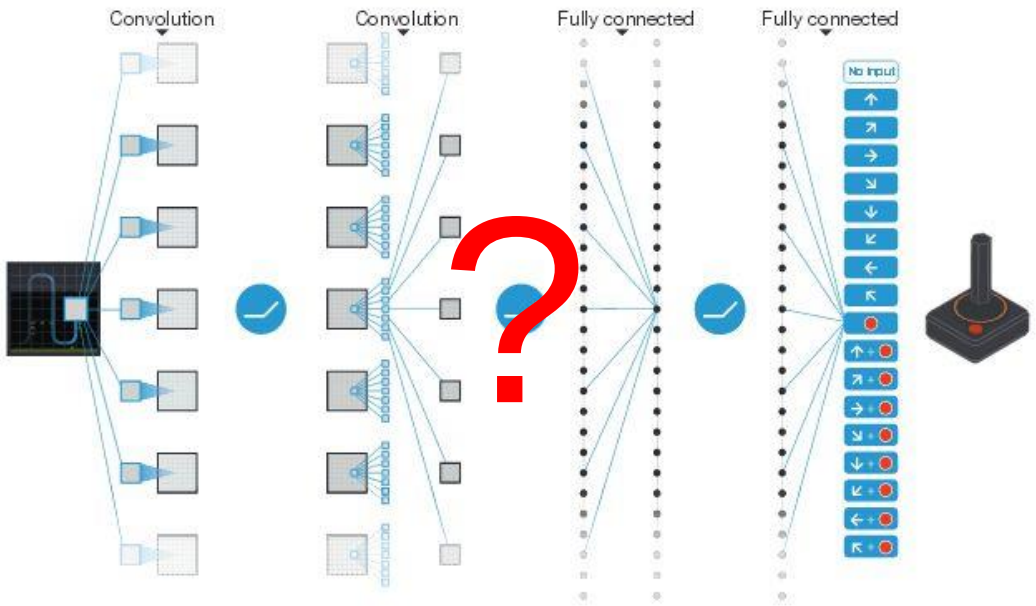
❑ 문제점

- 현실세계에서는 모든 state들이 정수, 자연수로 표현되지 않는다.
- 정수, 자연수로 표현되더라도 거대한 차원의 행렬을 필요로 한다.
- 예) CartPole => [a,b,c,d], Breakout => [210, 180, 3]
- 행동이 Discrete하게 정의되지 않을 수 있다.
- 예) 자율주행 자동차 => 휠의 각도, 제동의 세기, 가속의 세기

해결책

DeepMind

❑ 뉴럴넷으로 해결



해결책

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

Abstract

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

The theory of reinforcement learning provides a normative account[1], deeply rooted in psychological[2] and neuroscientific[3] perspectives on animal behaviour, of how agents may optimize their control of an environment. To use reinforcement learning successfully in situations approaching real-world complexity, however, agents are confronted with a difficult task: they must derive efficient representations of the environment from high-dimensional sensory inputs, and use these to generalize past experience to new situations. Remarkably, humans and other animals seem to solve this problem through a harmonious combination of reinforcement learning and hierarchical sensory processing systems[4,5], the former evidenced by a wealth of neural data revealing notable parallels between the phasic signals emitted by dopaminergic neurons and temporal difference reinforcement learning algorithms[3]. While reinforcement learning agents have achieved some successes in a variety of domains[6–8], their applicability has previously been limited to domains in which useful features can be handcrafted, or to domains with fully observed, low-dimensional state spaces. Here we use recent advances in training deep neural networks[9–11] to develop a novel artificial agent, termed a deep Q-network, that can learn successful policies directly from high-dimensional sensory inputs using end-to-end reinforcement learning. We tested this agent on the challenging domain of classic Atari 2600 games[12]. We demonstrate that the deep Q-network agent, receiving only the pixels and the game score as inputs, was able to surpass the performance of all previous algorithms and achieve a level comparable to that of a professional human games tester across a set of 49 games, using the same algorithm, network architecture and hyperparameters. This work bridges the divide between high-dimensional sensory inputs and actions, resulting in the first artificial agent that is capable of learning to excel at a diverse array of challenging tasks.

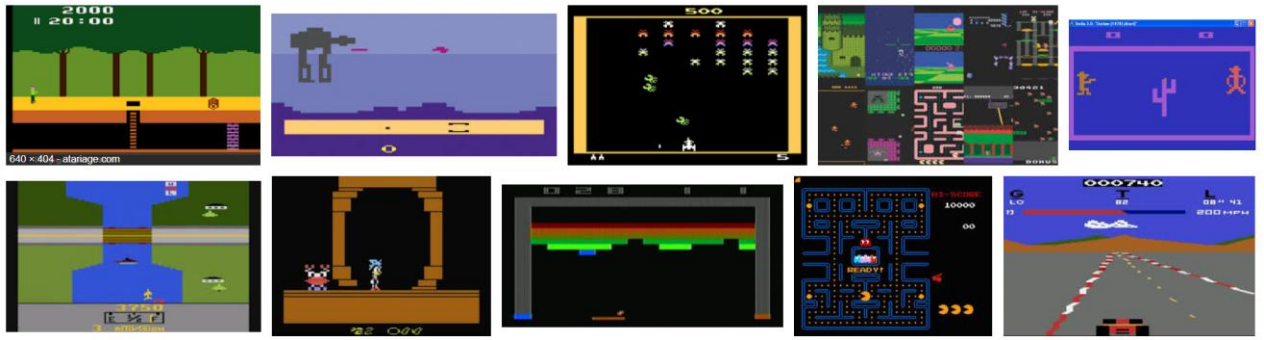
We set out to create a single algorithm that would be able to develop a wide range of competencies on a varied range of challenging tasks—a agent is to select actions in a fashion that maximizes cumulative future reward. More formally, we use a deep convolutional neural network to approximate the optimal action-value function

$$Q^*(s,a) = \max_{\pi} \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \mid s_t = s,\, a_t = a,\, \pi\right],$$

which is the maximum sum of rewards r_t discounted by γ at each time-step t, achievable by a behaviour policy $\pi = P(a|s)$, after making an observation (s) and taking an action (a) (see Methods)[19].

Reinforcement learning is known to be unstable or even to diverge when a nonlinear function approximator such as a neural network is used to represent the action-value (also known as Q) function[20]. This instability has several causes: the correlations present in the sequence of observations, the fact that small updates to Q may significantly change the policy and therefore change the data distribution, and the correlations between the action-values (Q) and the target values $r + \gamma \max_{a'} Q(s', a')$. We address these instabilities with a novel variant of Q-learning, which uses two key ideas. First, we used a biologically inspired mechanism termed experience replay[21–23] that randomizes over the data, thereby removing correlations in the observation sequence and smoothing over changes in the data distribution (see below for details). Second, we used an iterative update that adjusts the action-values (Q) towards target values that are only periodically updated, thereby reducing correlations with the target.

While other stable methods exist for training neural networks in the reinforcement learning setting, such as neural fitted Q-iteration[24], these methods involve the repeated training of networks *de novo* on hundreds of iterations. Consequently, these methods, unlike our algorithm, are too inefficient to be used successfully with large neural networks. We parameterize an approximate value function $Q(s,a;\theta_i)$ using the deep convolutional neural network shown in Fig. 1, in which θ_i are the parameters (that is, weights) of the Q-network at iteration i. To perform experience replay we store the agent's experiences $e_t = (s_t, a_t, r_t, s_{t+1})$

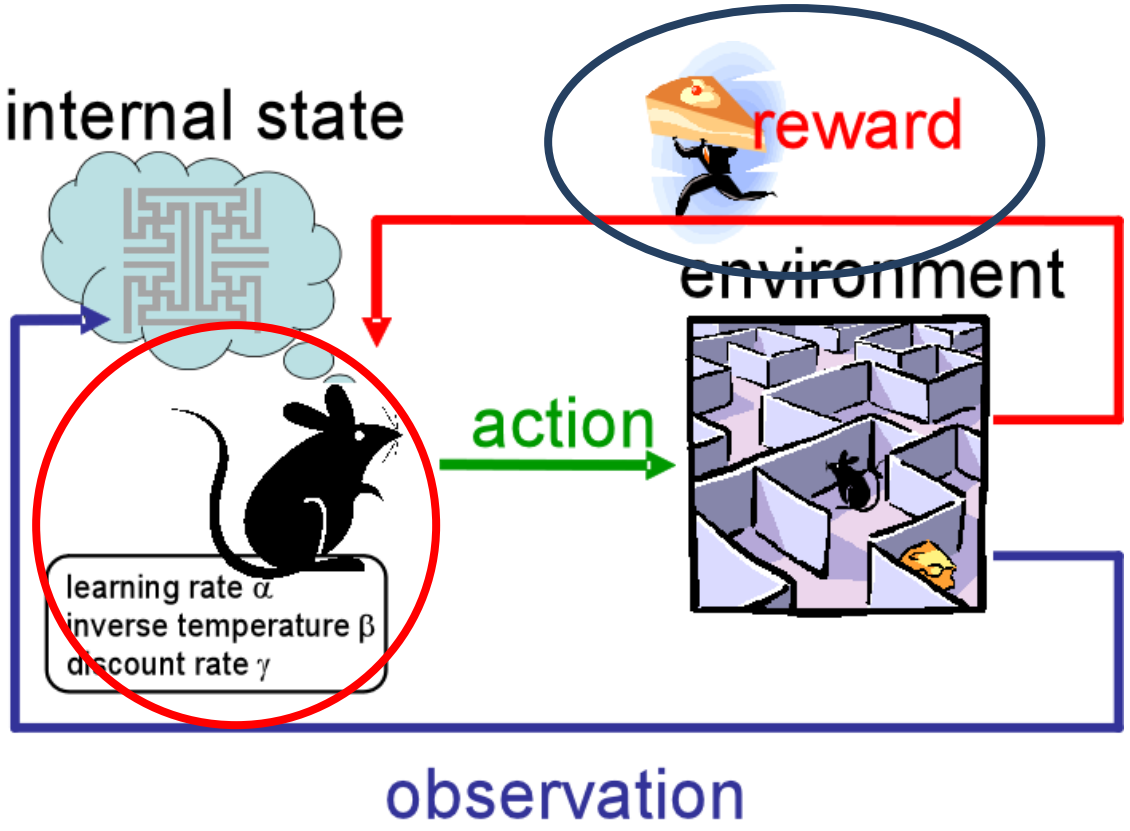


강화학습 연구분야

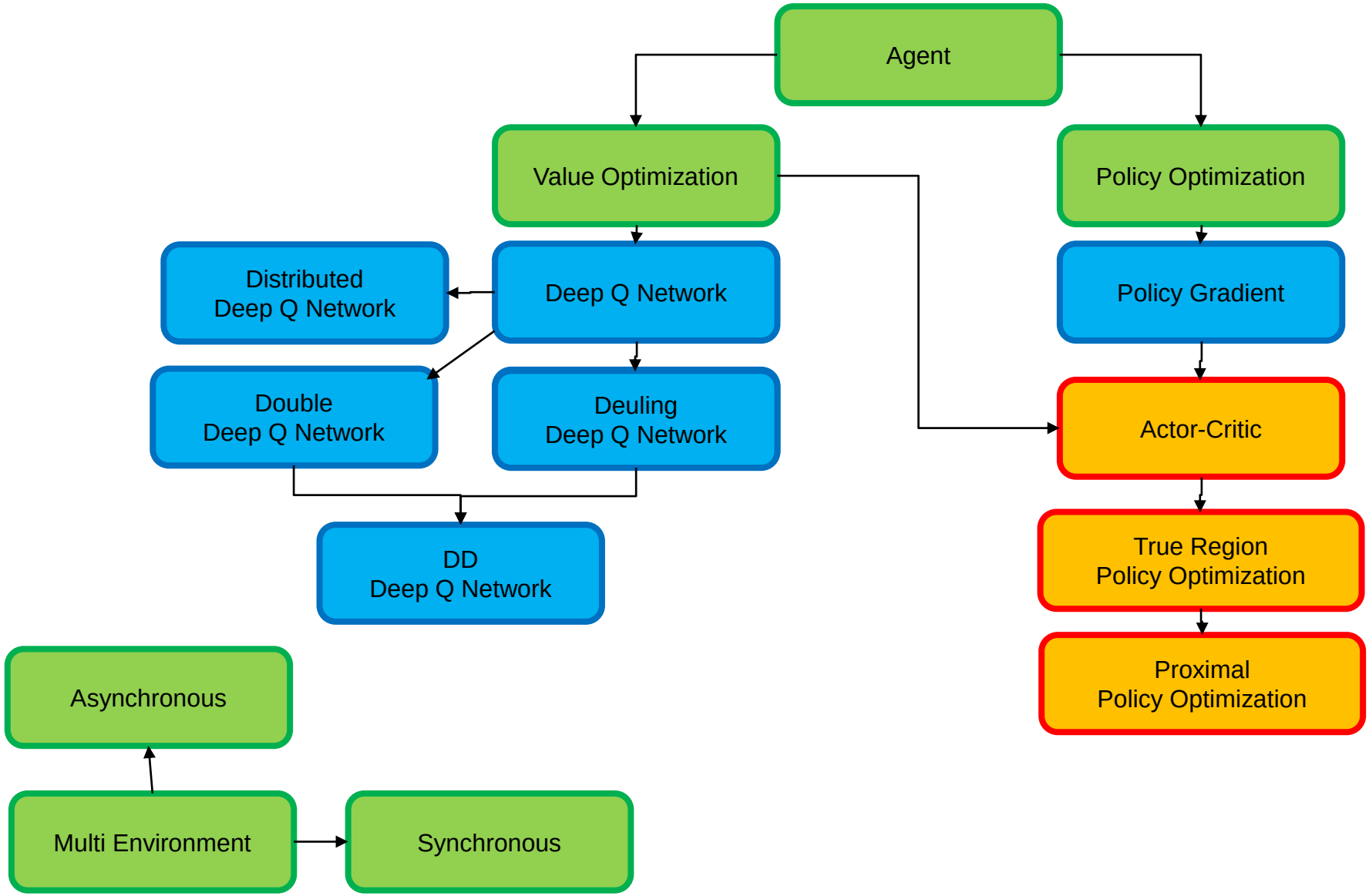
Idea Suggestions

KeumGang Cha

연구 분야


internal state
reward
environment
action
learning rate α
inverse temperature β
discount rate γ
observation

알고리즘


Agent
Value Optimization
Policy Optimization
Distributed
Deep Q Network
Deep Q Network
Policy Gradient
Double
Deep Q Network
Deuling
Deep Q Network
Actor-Critic
DD
Deep Q Network
True Region
Policy Optimization
Proximal
Policy Optimization
Asynchronous
Multi Environment
Synchronous

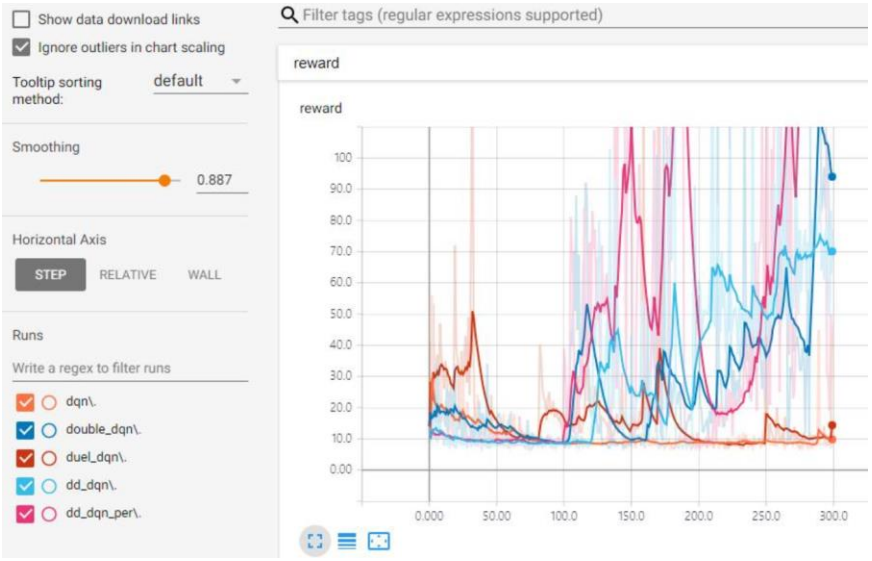
성능

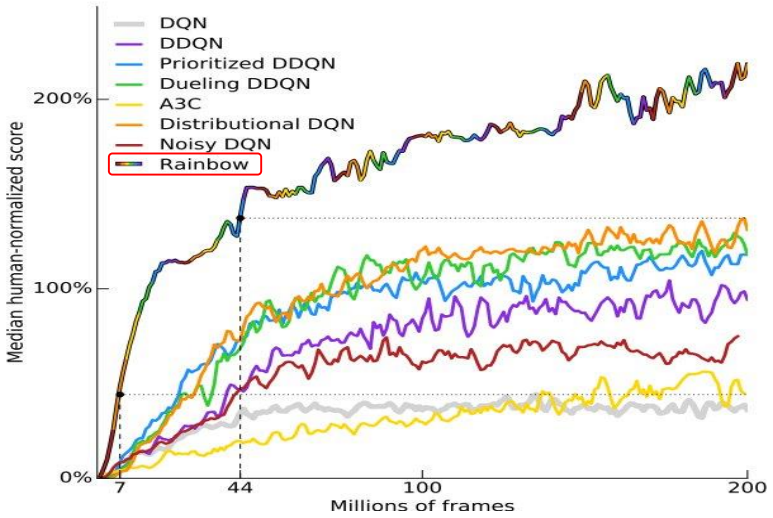
❑ Value Optimization

- 성능 비교는 Value Optimization끼리만 비교가 가능
- 네트워크 구성 자체가 다르기 때문

❑ Policy-Value Optimization

- OpenAI, ML-Unity 등 많은 강화학습 관련 회사에서 Proximal Policy Optimization(PPO)를 Baseline으로 사용





보상

❑ GAE(Generalized Advantaged Estimation)

High-Dimensional Continuous Control Using Generalized Advantage Estimation

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, Pieter Abbeel

(Submitted on 8 Jun 2015 (v1), last revised 9 Sep 2016 (this version, v5))

Policy gradient methods are an appealing approach in reinforcement learning because they directly optimize the cumulative reward and can straightforwardly be used with nonlinear function approximators such as neural networks. The two main challenges are the large number of samples typically required, and the difficulty of obtaining stable and steady improvement despite the nonstationarity of the incoming data. We address the first challenge by using value functions to substantially reduce the variance of policy gradient estimates at the cost of some bias, with an exponentially-weighted estimator of the advantage function that is analogous to TD(lambda). We address the second challenge by using trust region optimization procedure for both the policy and the value function, which are represented by neural networks.
Our approach yields strong empirical results on highly challenging 3D locomotion tasks, learning running gaits for bipedal and quadrupedal simulated robots, and learning a policy for getting the biped to stand up from starting out lying on the ground. In contrast to a body of prior work that uses hand-crafted policy representations, our neural network policies map directly from raw kinematics to joint torques. Our algorithm is fully model-free, and the amount of simulated experience required for the learning tasks on 3D bipeds corresponds to 1-2 weeks of real time.

❑ HER(Hindsight Experience Replay)

Hindsight Experience Replay

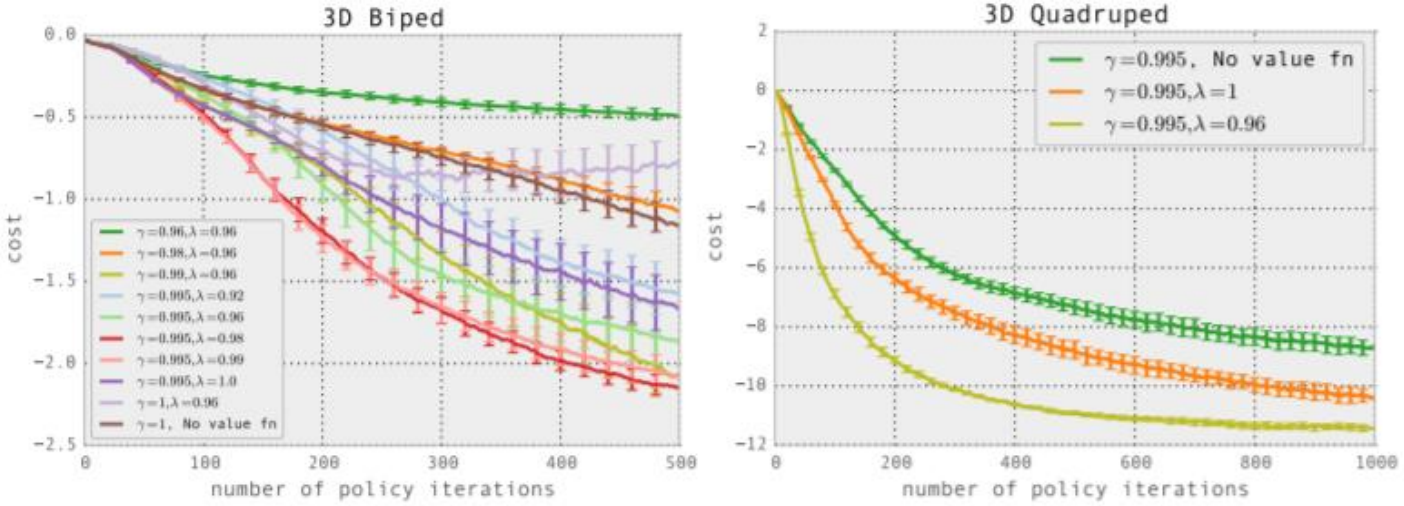
Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, Wojciech Zaremba

(Submitted on 5 Jul 2017 (v1), last revised 23 Feb 2018 (this version, v3))

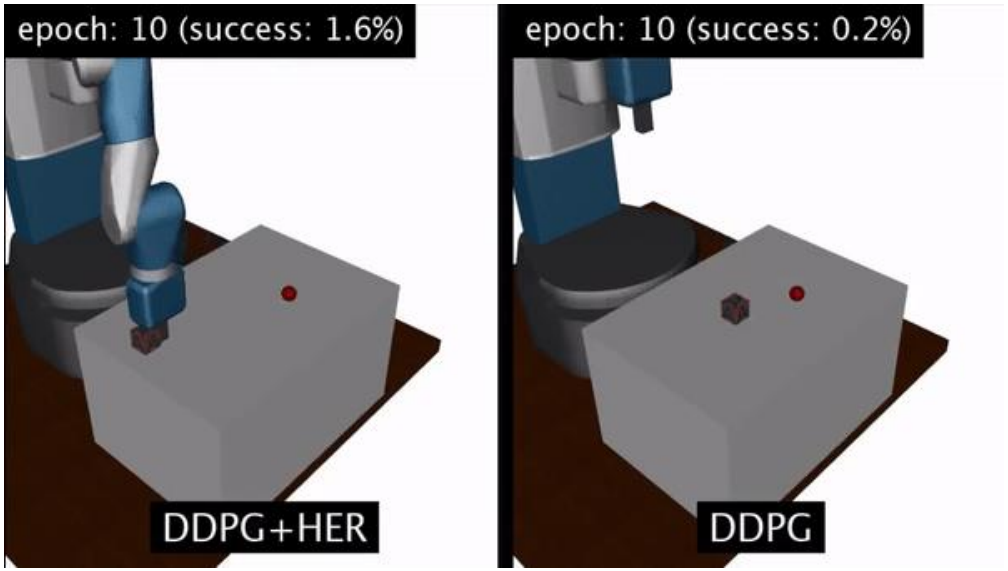
Dealing with sparse rewards is one of the biggest challenges in Reinforcement Learning (RL). We present a novel technique called Hindsight Experience Replay which allows sample-efficient learning from rewards which are sparse and binary and therefore avoid the need for complicated reward engineering. It can be combined with an arbitrary off-policy RL algorithm and may be seen as a form of implicit curriculum.
We demonstrate our approach on the task of manipulating objects with a robotic arm. In particular, we run experiments on three different tasks: pushing, sliding, and pick-and-place, in each case using only binary rewards indicating whether or not the task is completed. Our ablation studies show that Hindsight Experience Replay is a crucial ingredient which makes training possible in these challenging environments. We show that our policies trained on a physics simulation can be deployed on a physical robot and successfully complete the task.

성능

- **GAE**


3D Biped
cost
number of policy iterations
γ=0.96,λ=0.96
γ=0.98,λ=0.96
γ=0.99,λ=0.96
γ=0.995,λ=0.92
γ=0.995,λ=0.96
γ=0.995,λ=0.98
γ=0.995,λ=0.99
γ=0.995,λ=1.0
γ=1,λ=0.96
γ=1, No value fn
3D Quadruped
cost
number of policy iterations
γ=0.995, No value fn
γ=0.995,λ=1
γ=0.995,λ=0.96

- **HER**


epoch: 10 (success: 1.6%)
DDPG+HER
epoch: 10 (success: 0.2%)
DDPG

환경

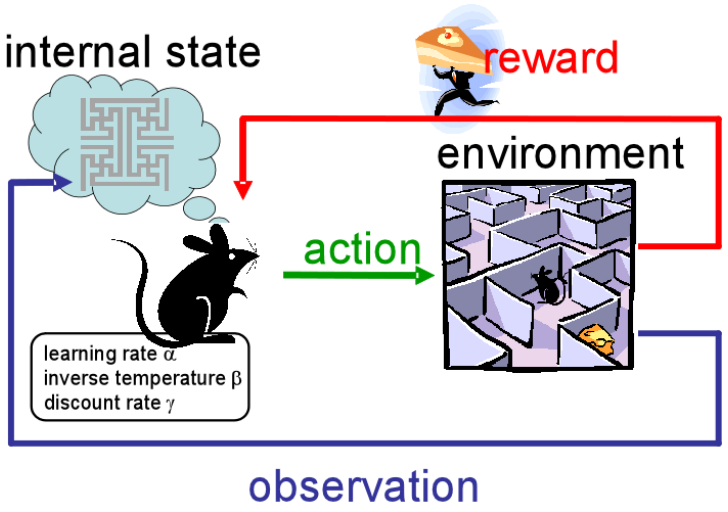
Idea Suggestions

KeumGang Cha

환경이란

❑ 환경의 조건

- Agent와의 상호작용이 가능
- Agent의 Action에 따른 환경의 변화
- Agent의 Action에 따른 보상
- Agent의 목표 미달성에 따른 손실 제거

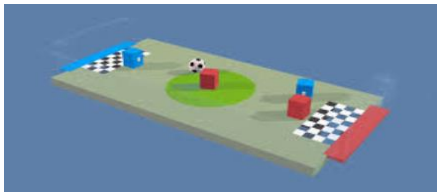


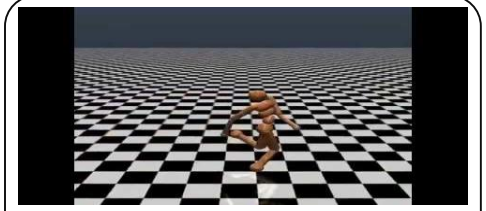
❑ 최적의 환경(시뮬레이션)

- 게임(인간 지능이 가장 많이 필요한 분야)
- 스타크래프트, 바둑, 체스, Atari, 카드게임

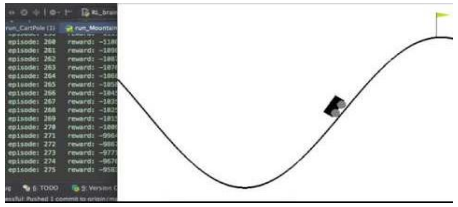
무료로 제공되는 환경

unity





OpenAI



deep drive

DeepMind

BLIZZARD ENTERTAINMENT

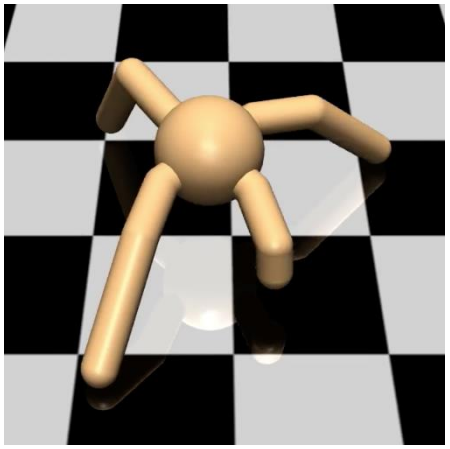
어떤 환경으로 강화학습을 할까

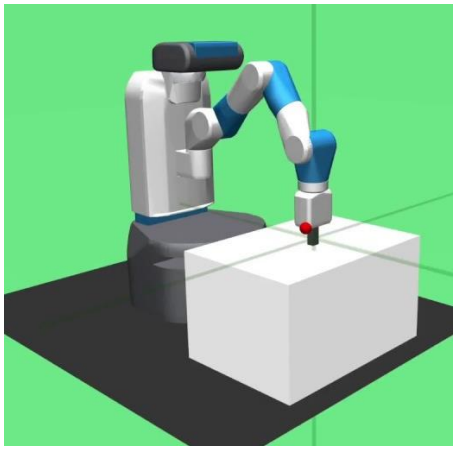
Idea Suggestions

KeumGang Cha

여러가지 환경

❑ 로보틱스





❑ 시뮬레이션

무엇을 선택해야할까

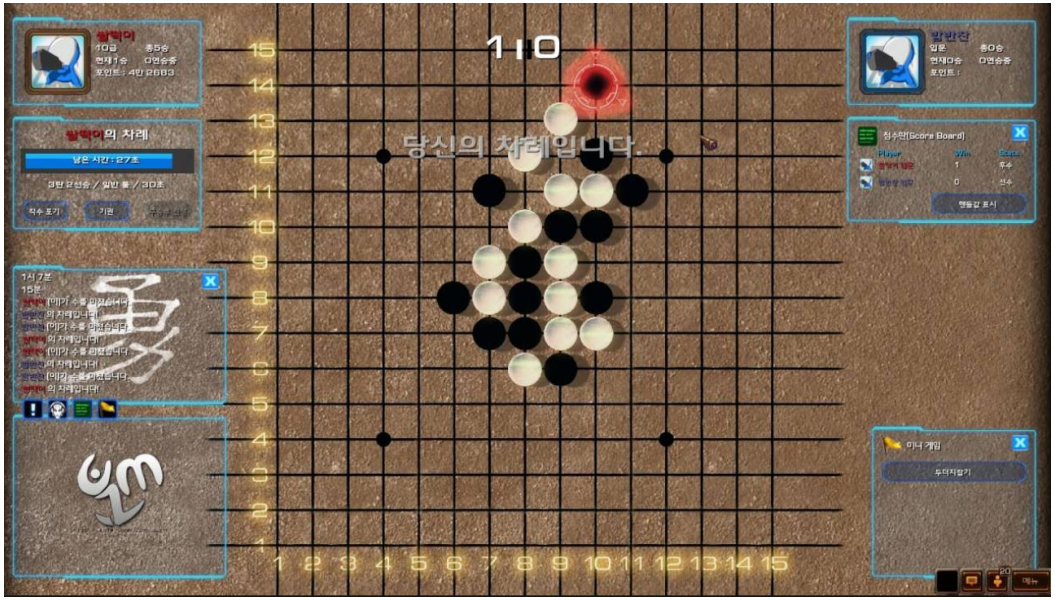
왜 스타크래프트2인가?

'스타크래프트 2'가 11월 14일부터 무료화된다

김태우
허핑턴포스트코리아

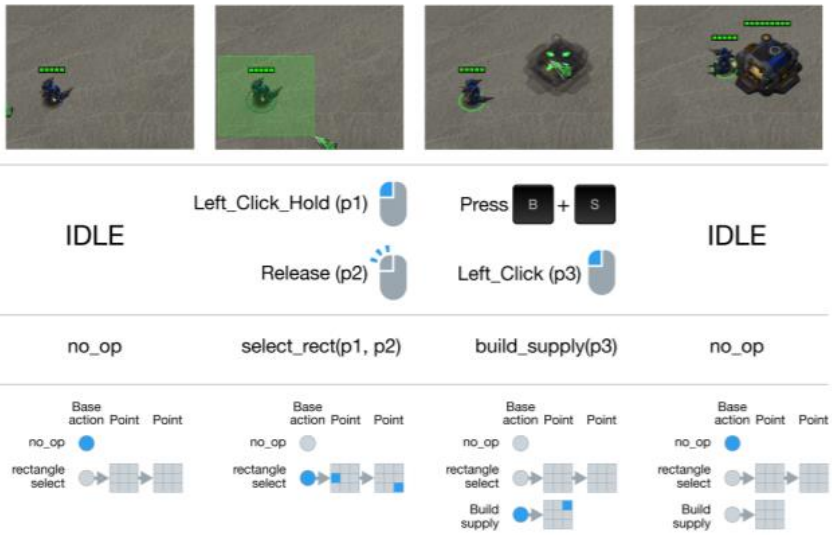
블리자드가 '스타크래프트 2'의 무료화를 선언했다.

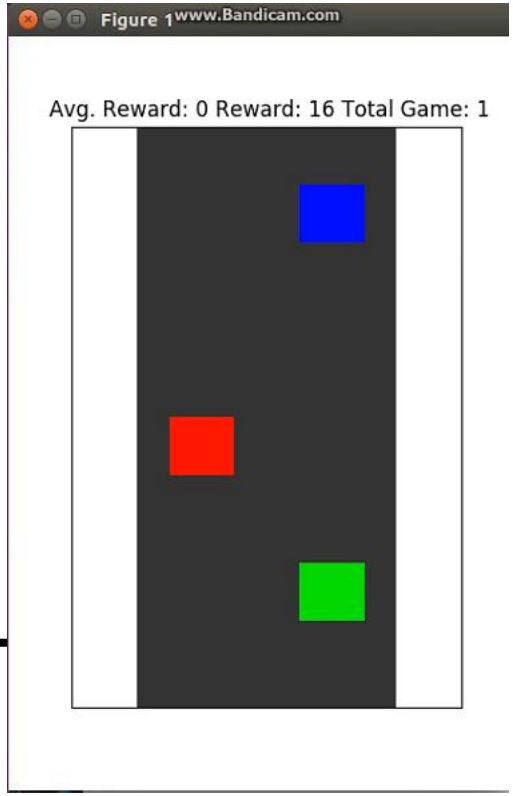
더 버지에 따르면 블리자드 엔터테인먼트는 지난 3일, 블리즈컨 2017 행사에서 '스타크래프트 2'의 일부 콘텐츠를 무료로 배포하겠다고 발표했다. 무료화되는 콘텐츠는 '자유의 날개' 캠페인과 멀티 플레이어 게임, 모든 협동전 사령관 등이다. 유저들은 오는 14일부터 게임을 무료로 다운받을 수 있으며, 나머지 캠페인은 개별 구매를 통해 즐길 수 있다.



왜 스타크래프트2인가?


```
Function.ability(507, "TrainWarp_HighTemplar_screen", cmd_screen, 1416),
Function.ability(508, "TrainWarp_Sentry_screen", cmd_screen, 1418),
Function.ability(509, "TrainWarp_Stalker_screen", cmd_screen, 1414),
Function.ability(510, "TrainWarp_Zealot_screen", cmd_screen, 1413),
Function.ability(511, "UnloadAll_quick", cmd_quick, 3664),
Function.ability(512, "UnloadAll_Bunker_quick", cmd_quick, 408, 3664),
Function.ability(513, "UnloadAll_CommandCenter_quick", cmd_quick, 413, 3664),
Function.ability(514, "UnloadAll_NydasNetwork_quick", cmd_quick, 1438, 3664),
Function.ability(515, "UnloadAll_NydusWorm_quick", cmd_quick, 2371, 3664),
Function.ability(516, "UnloadAllAt_screen", cmd_screen, 3669),
Function.ability(517, "UnloadAllAt_minimap", cmd_minimap, 3669),
Function.ability(518, "UnloadAllAt_Medivac_screen", cmd_screen, 396, 3669),
Function.ability(519, "UnloadAllAt_Medivac_minimap", cmd_minimap, 396, 3669),
Function.ability(520, "UnloadAllAt_Overlord_screen", cmd_screen, 1408, 3669),
Function.ability(521, "UnloadAllAt_Overlord_minimap", cmd_minimap, 1408, 3669),
Function.ability(522, "UnloadAllAt_WarpPrism_screen", cmd_screen, 913, 3669),
Function.ability(523, "UnloadAllAt_WarpPrism_minimap", cmd_minimap, 913, 3669),
```





THANK YOU

Any questions or comments?

E-mail : chagmgang@gmail.com, chagmgang@plani.co.kr

Github : http://github.com/chagmgang