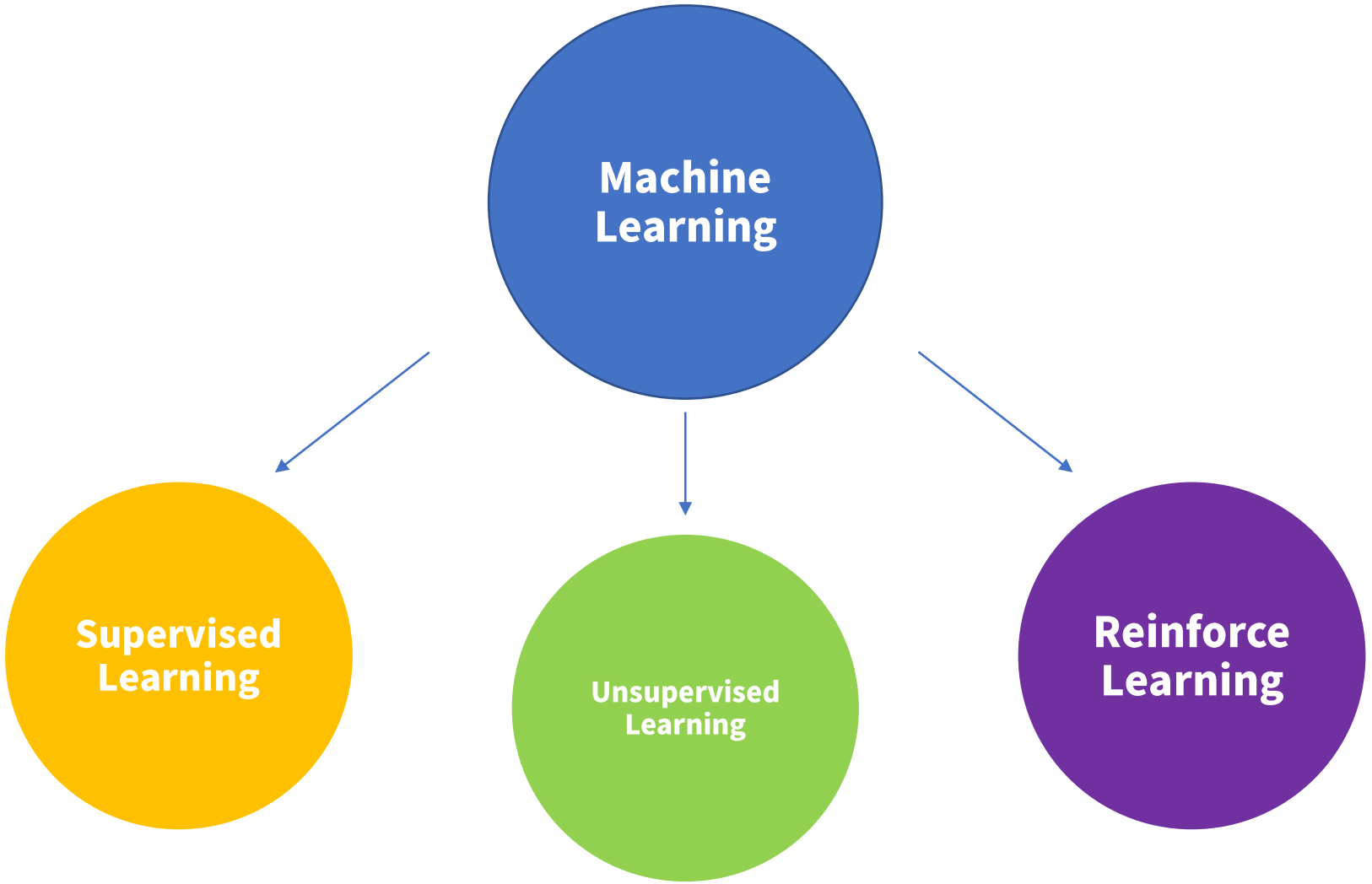
Snake Game in Reinforce Learning

20201 Joseph Kang
20212 Ted Song

Index

- What is Reinforce learning
- What is Q learning
- What is Deep Q learning (DQN)
- Difference between Q learning and DQN
- Snake Game with Reinforce learning
  - Define Sensors Architecture
  - Define Layers Architecture
  - Fitness
  - Limitations
- Concluding…

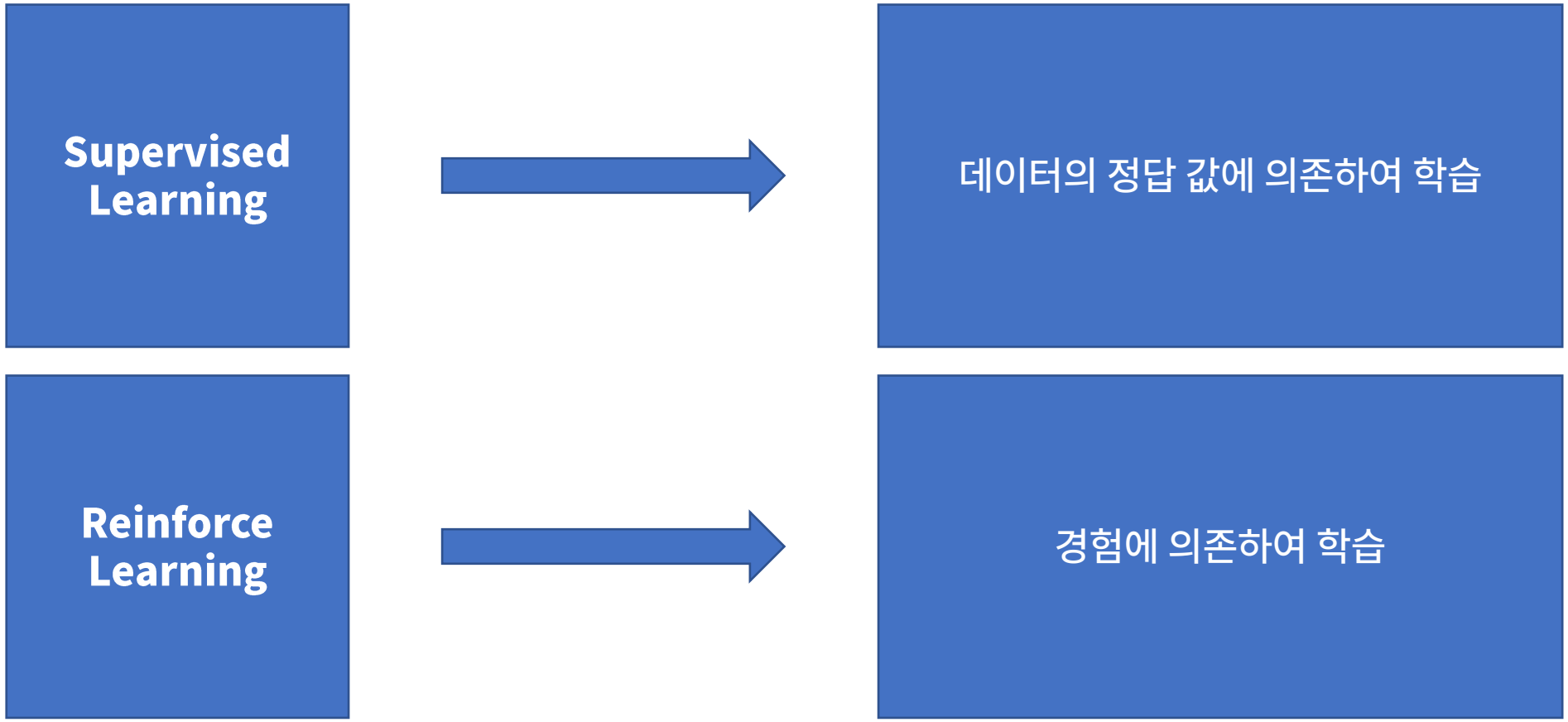
What is Reinforce learning



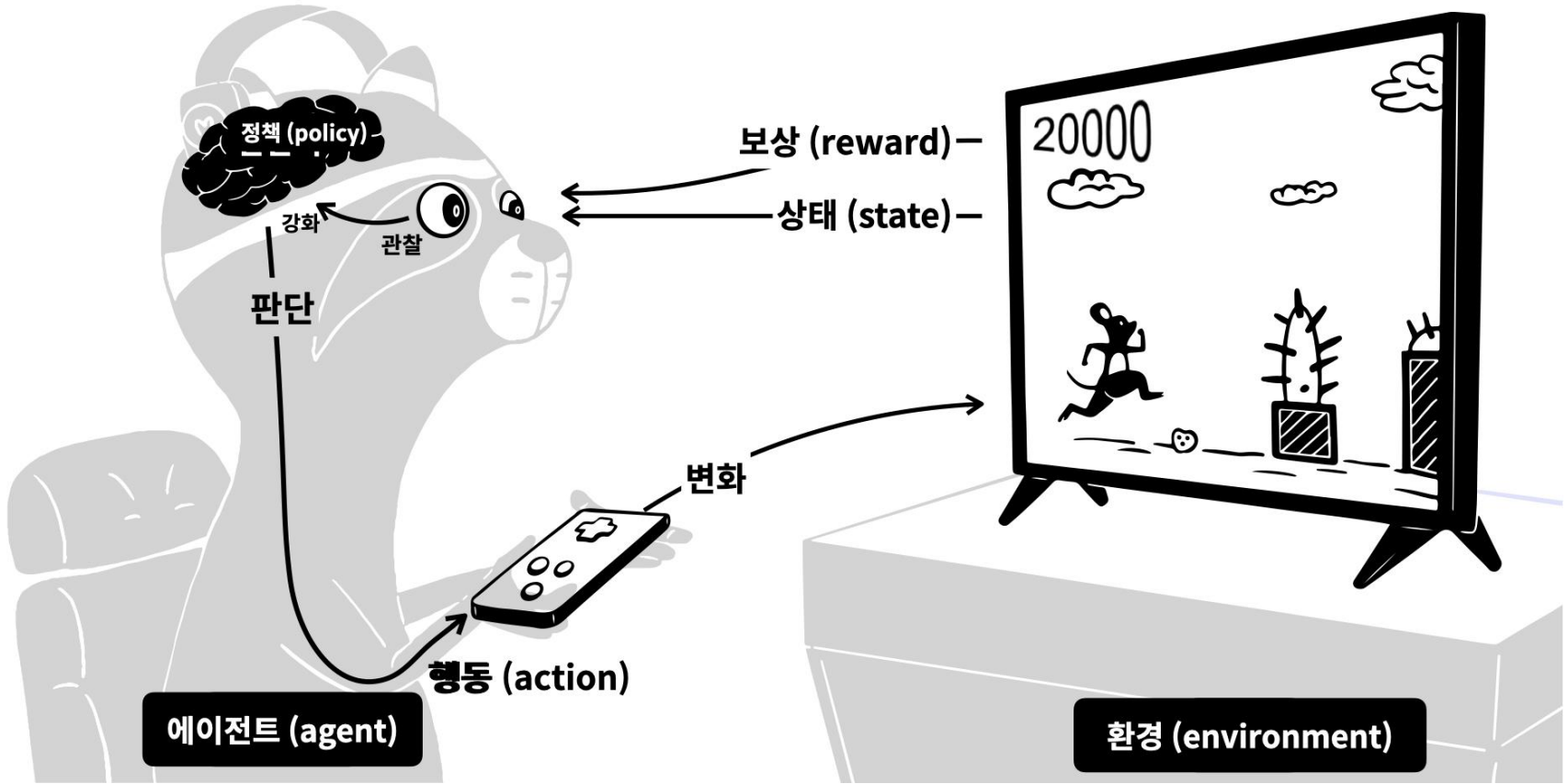
What is Reinforce learning

Action > N ? Reward : Penalty

What is Reinforce learning

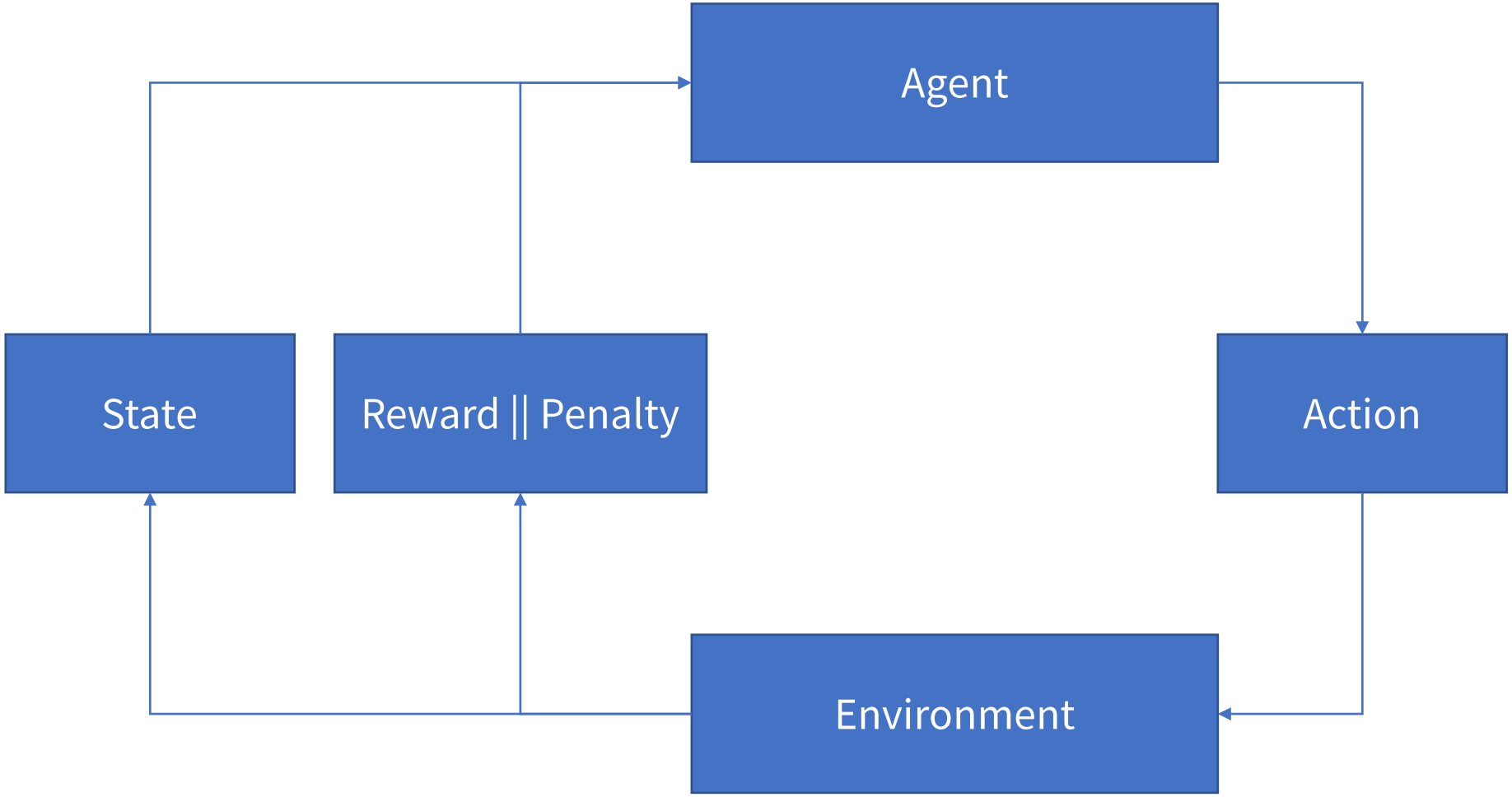


What is Reinforce learning



IMG : 생활코딩

| What is Reinforce learning



| What is Q learning

Q-value가 저장된 **Q-table**의 값을
Action과 State별로 최고의 선택을 하는 것

What is Q learning

< The Bellman Equation >

$$Q(state, action) = R + \max\left(Q(state', action)\right)$$

$$Q(state, action) = R + \gamma * \max(Q(state', action))$$

What is Q learning

1. Random Noise

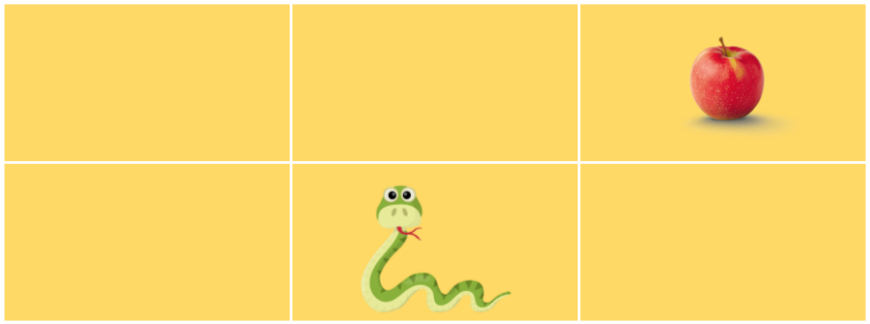
- 현재 State에서 가능한 Action에 따른 Q값에 Random으로 Noise값을 주고, 이것이 최대 값을 가지는 Action을 선택하는 것
- 무조건 최대 Q값을 따르지 않고, 가끔은 다른 Action을 선택한다.

2. ε-greedy

- 임의의 확률 값 e를 주고, e의 확률로 탐색한다.
- 예를 들어 $e = 0.99$ 라면 99%의 확률로 최대 Q값으로 탐색하고, 1%의 확률로 새로운 길을 찾는다.

What is Q learning

Game Board



State (s) : $\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$

Selected Motion (a) : "Right"

Reward (r) :

Next State (s') :

$\max Q(s')$:

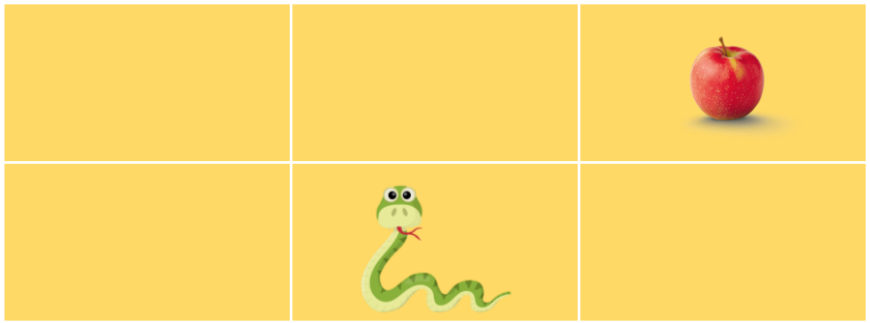
Q Table

$discount = 0.95$

	$\begin{bmatrix}0&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&0\end{bmatrix}$
Up	0.2	0.3	1.0	-0.22	-0.3	0.0
Down	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
Right	0.21	0.4	-0.3	0.5	1.0	0.0

What is Q learning

Game Board



State (s) : $\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$

Selected Motion (a) : "Right"

Reward (r) : 0

Next State (s') :

$\max Q(s')$:

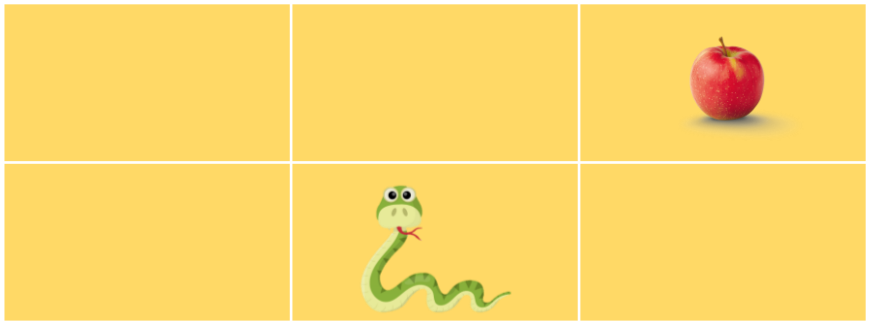
Q Table

$discount = 0.95$

	$\begin{bmatrix}0&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&0\end{bmatrix}$
Up	0.2	0.3	1.0	-0.22	-0.3	0.0
Down	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
Right	0.21	0.4	-0.3	0.5	1.0	0.0

What is Q learning

Game Board



State (s) : $\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$

Selected Motion (a) : "Right"

Reward (r) : 0

Next State (s') : $\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$

$\max Q(s')$:

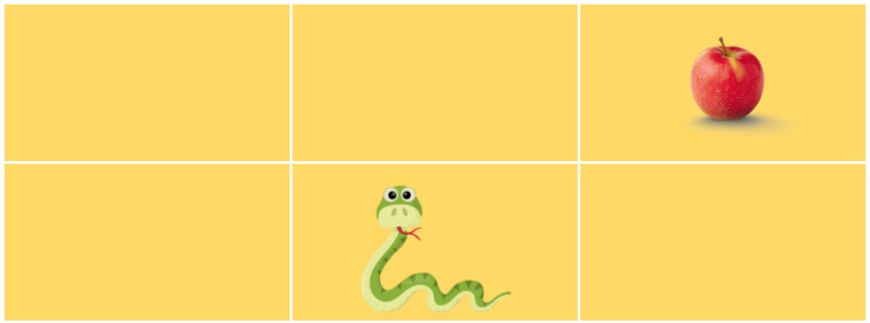
Q Table

$discount = 0.95$

	$\begin{bmatrix}0&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&0\end{bmatrix}$
Up	0.2	0.3	1.0	-0.22	-0.3	0.0
Down	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
Right	0.21	0.4	-0.3	0.5	1.0	0.0

What is Q learning

Game Board



State (s) : $\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$

Selected Motion (a) : "Right"

Reward (r) : 0

Next State (s') : $\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$

$\max Q(s')$: 1.0

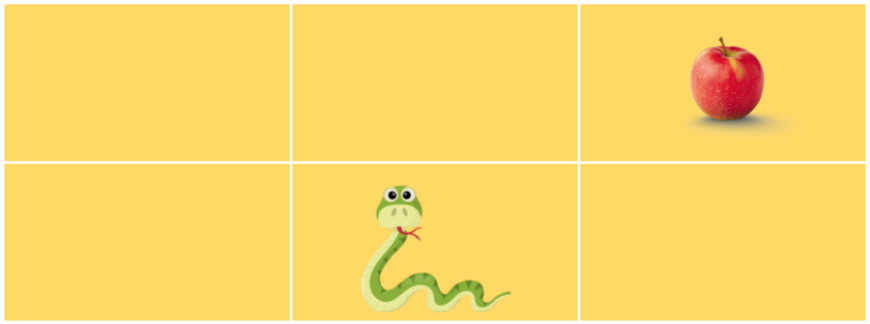
Q Table

$discount = 0.95$

	$\begin{bmatrix}0&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&0\end{bmatrix}$
Up	0.2	0.3	1.0	-0.22	-0.3	0.0
Down	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
Right	0.21	0.4	-0.3	0.5	1.0	0.0

What is Q learning

Game Board



Q Table

$discount = 0.95$

	$\begin{bmatrix}0&0&0\\1&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$	$\begin{bmatrix}1&0&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&1&0\\0&0&0\end{bmatrix}$	$\begin{bmatrix}0&0&1\\0&0&0\end{bmatrix}$
Up	0.2	0.3	1.0	-0.22	-0.3	0.0
Down	-0.5	-0.4	-0.2	-0.04	-0.02	0.0
Right	0.21	0.95	-0.3	0.5	1.0	0.0

State (s) : $\begin{bmatrix}0&0&0\\0&1&0\end{bmatrix}$

Selected Motion (a) : "Right"

Reward (r) : 0

Next State (s') : $\begin{bmatrix}0&0&0\\0&0&1\end{bmatrix}$

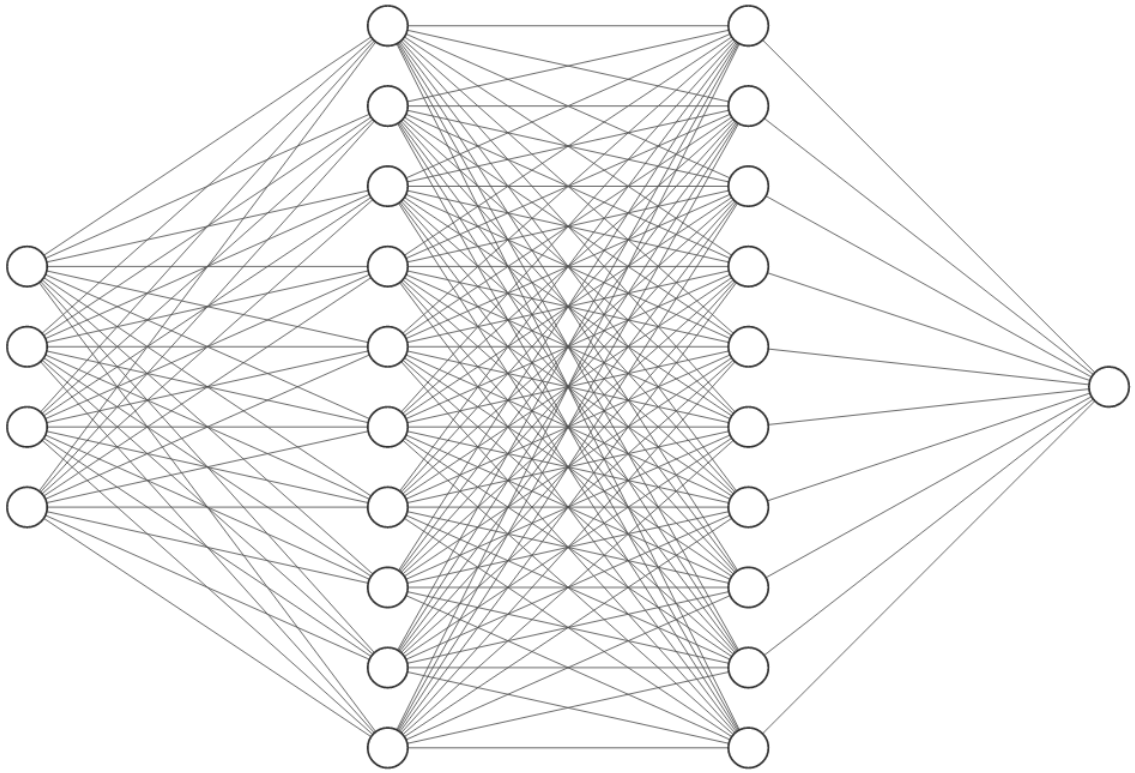
$\max_{a'} Q(s')$: 1.0

$$Q(s,a) := r + discount * max\, Q(s') = 0 + 0.95 * 1 = 0.95$$

What is DQN Learning

State-Action에 따른 Q-value를
Deep Learning을 통해서 구하는 방식

What is DQN Learning



Input Layer $\in \mathbb{R}^4$ Hidden Layer $\in \mathbb{R}^{10}$ Hidden Layer $\in \mathbb{R}^{10}$ Output Layer $\in \mathbb{R}^1$

- Q-Value 1
- Q-Value 2
- Q-Value 3
- Q-Value 4

What is DQN Learning

기존의 Q-Learning의 문제점

- State가 많아지면 많아질 수록 Table의 크기 또한 커진다.
- State가 만약 Color-Image라면, Memory 소모가 커진다.

기존의 Q-Learning의 문제점을 보완한 DQN

- Q-Value를 Table의 형태가 아닌, Weight를 가지는 Deep Learning의 형태로 설정한다.
- 하지만 강화학습에서 DQN 또한 문제점이 있다.

What is DQN Learning

Deep Q-Learning의 문제점

1. Deep Learning은 Label 값이 있는 Data를 학습시키는데, Reinforcement Learning은 Label이 없고, 가끔 들어오는 Reward로 학습을 시켜야 하기 때문에, 제대로 된 학습이 되기 힘들다.

2. Deep Learning은 Data Sample이 서로 독립적이라는 가정을 하지만, Reinforcement Learning에서는 다음 State가 현재 State과 연관성이 크기 때문에 이 가정이 성립하지 않는다.

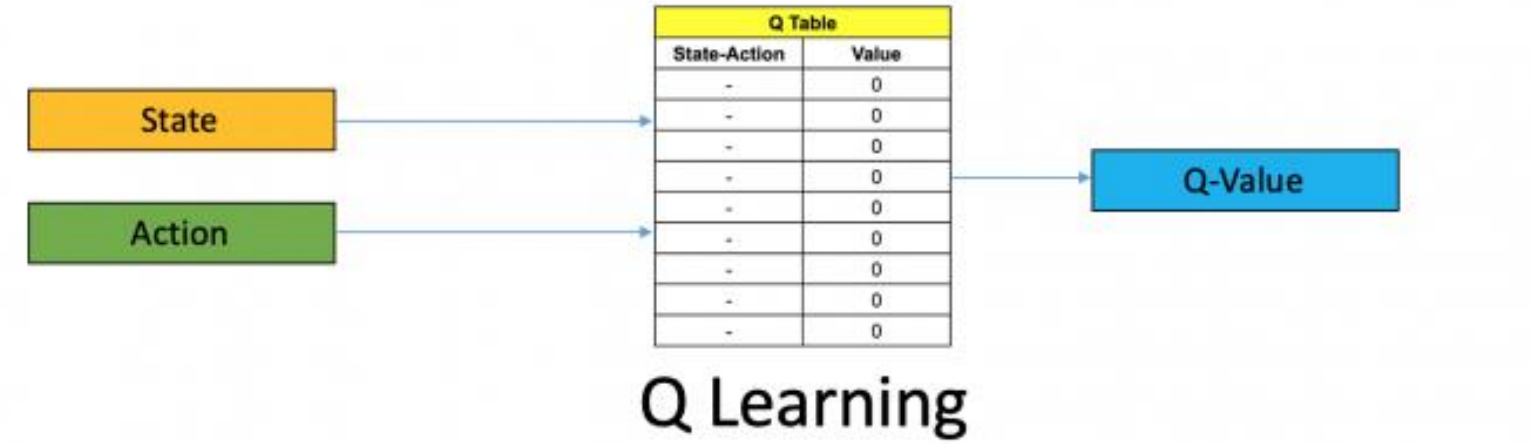
What is DQN Learning

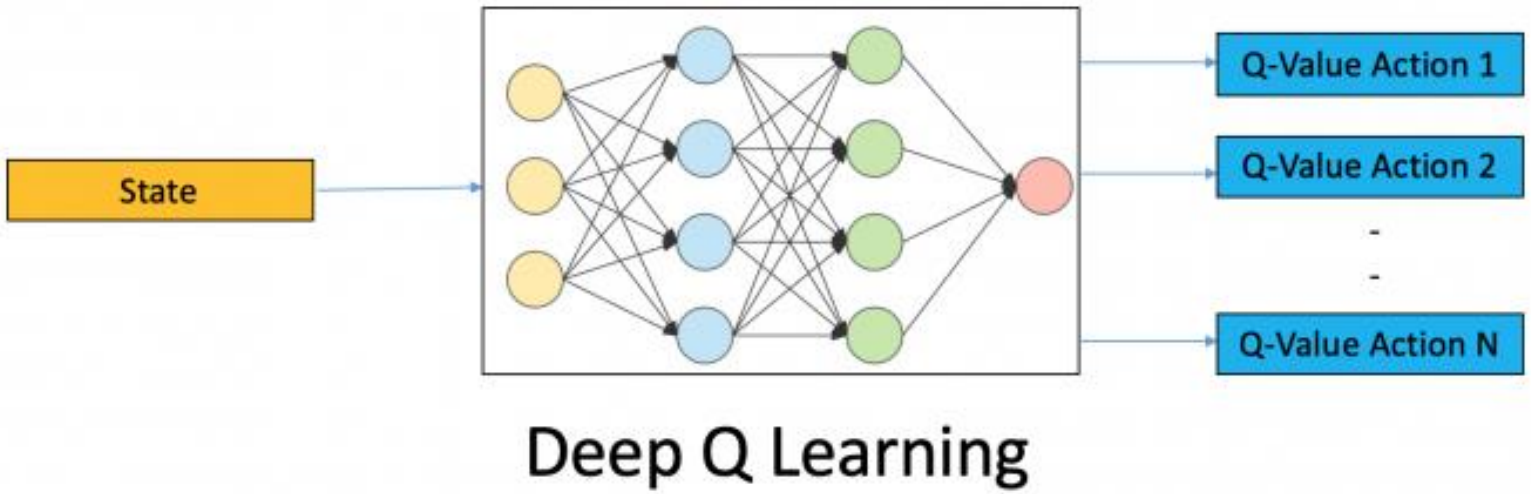
Deep Q-Learning의 문제를 해결해보자

강화학습 Episode를 진행하면서, 바로 Deep Learning의 Weight를 학습시키는 것이 아니라,
Time-Step마다 [S, A, R, S'] Data set을 모아서 학습하자는 방법

출처 논문 : Playing Atari with deep reinforcement learning. Technical Report arXiv:1312.5602 [cs.LG], Deepmind Technologies.12 (Mnih, V.)

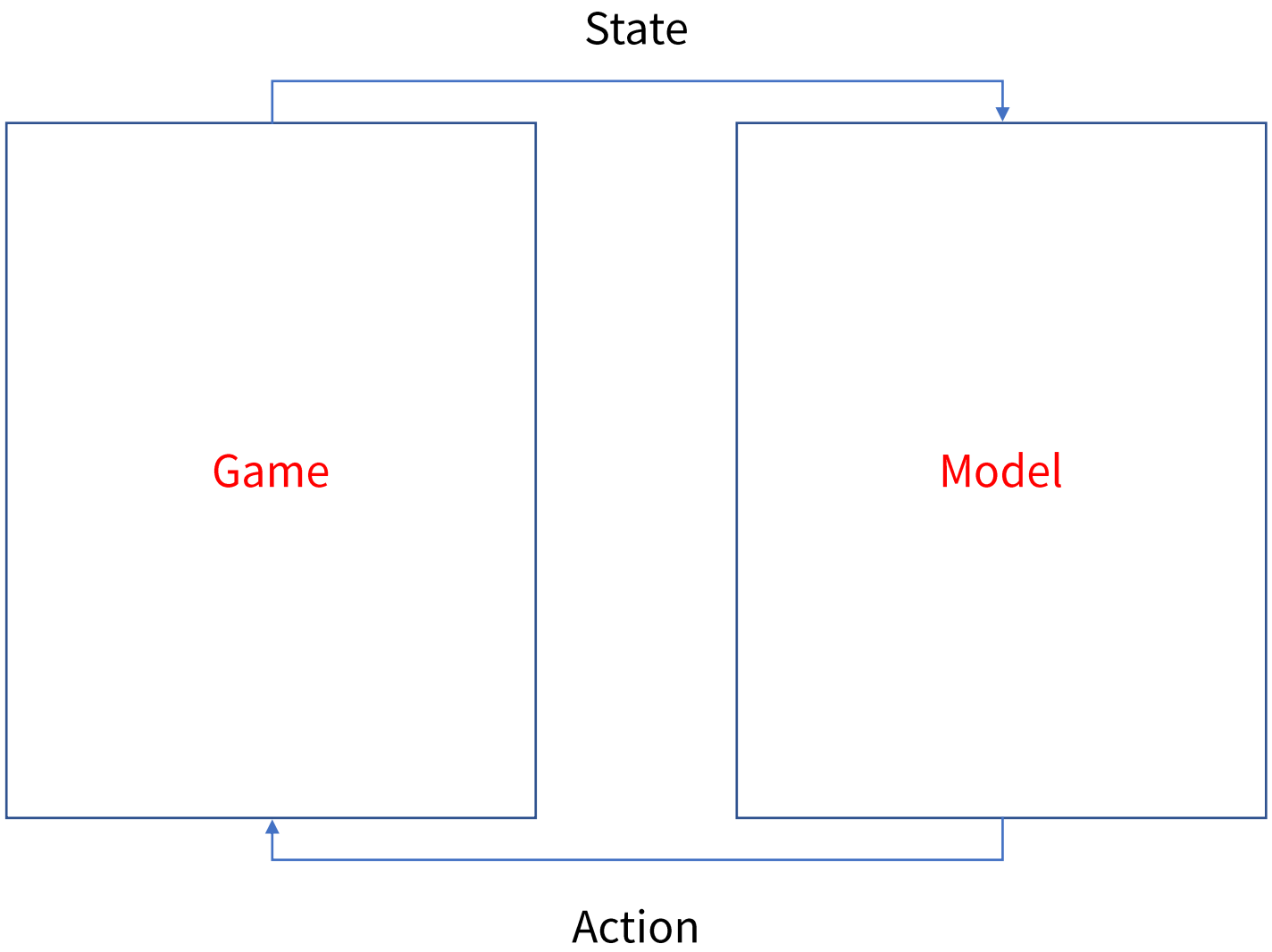
| Difference between Q learning and DQN





Snake Game with Reinforce learning

| Define Logic



Snake Game with Reinforce learning

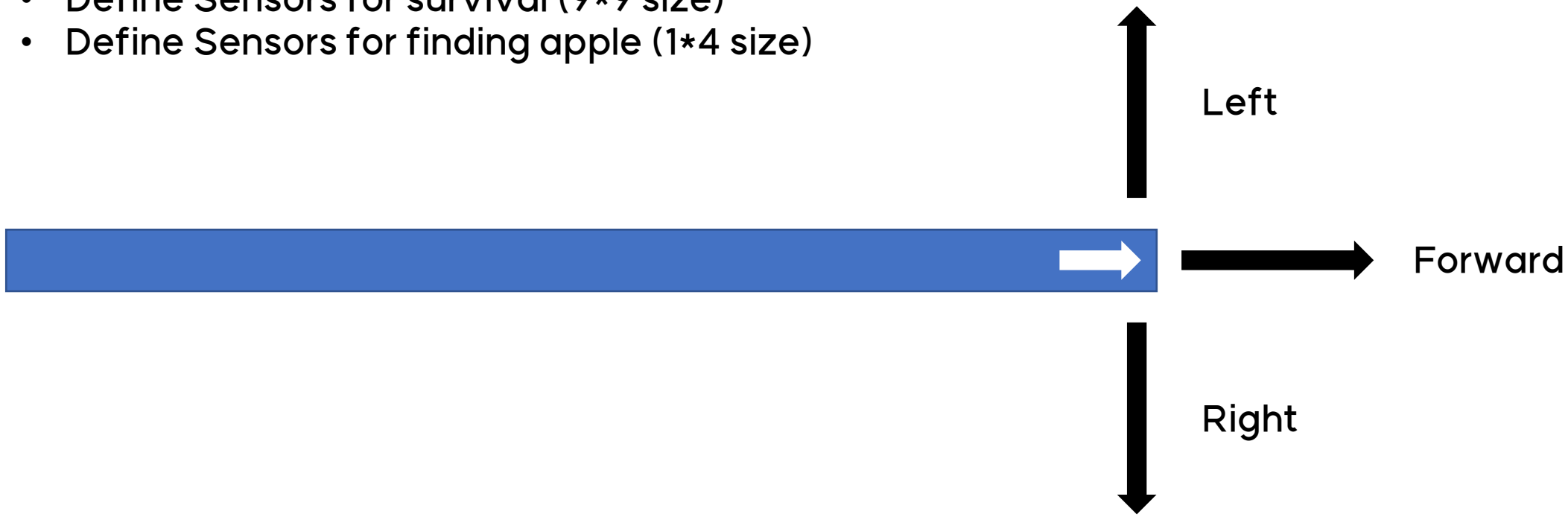
| Define Logic

학습 목표 : 효율적인 움직임으로 오래 Snake를 살 찌우자

1. init Q-Value (= init model)
2. Choose Action (model.select_action(state) or ε-greedy)
3. Perform Action
4. Measure Reward
5. Update Q-Value (+ optimize model)
6. Memorize reward for current action

Define Sensors Architecture

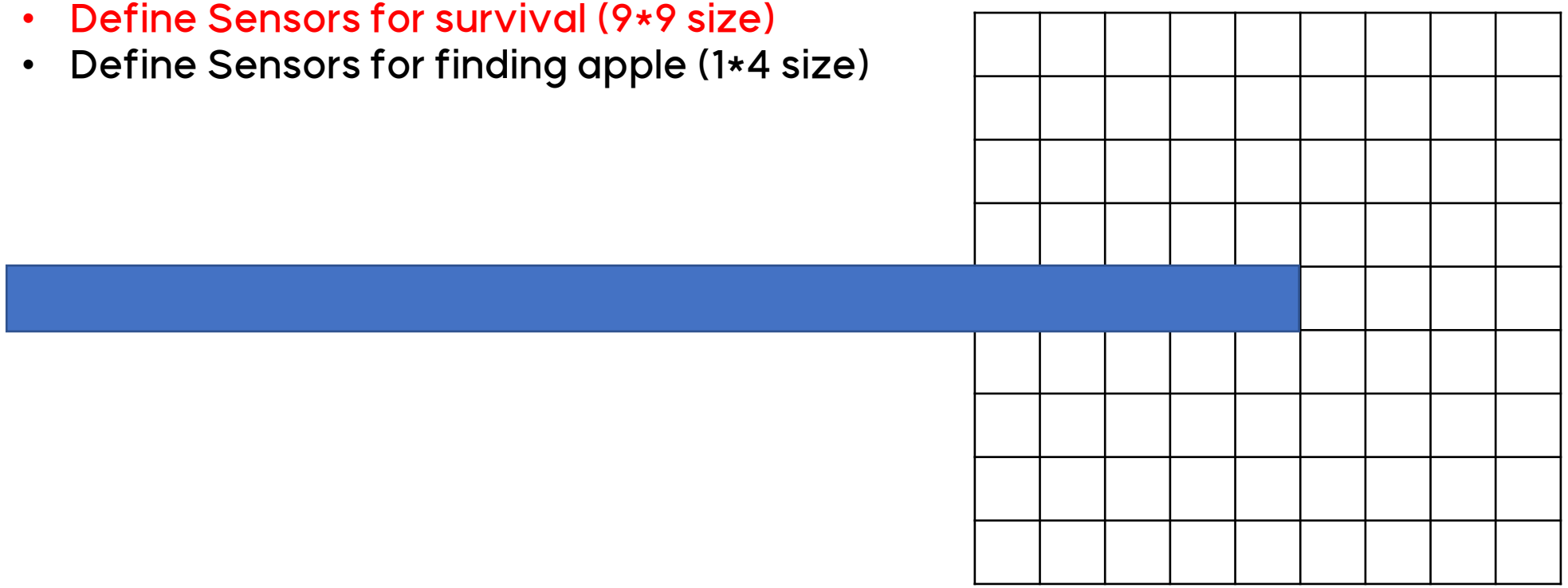
- Define Snake Movement
- Define Sensors for survival (9*9 size)
- Define Sensors for finding apple (1*4 size)



Snake Game with Reinforce learning

Define Sensors Architecture

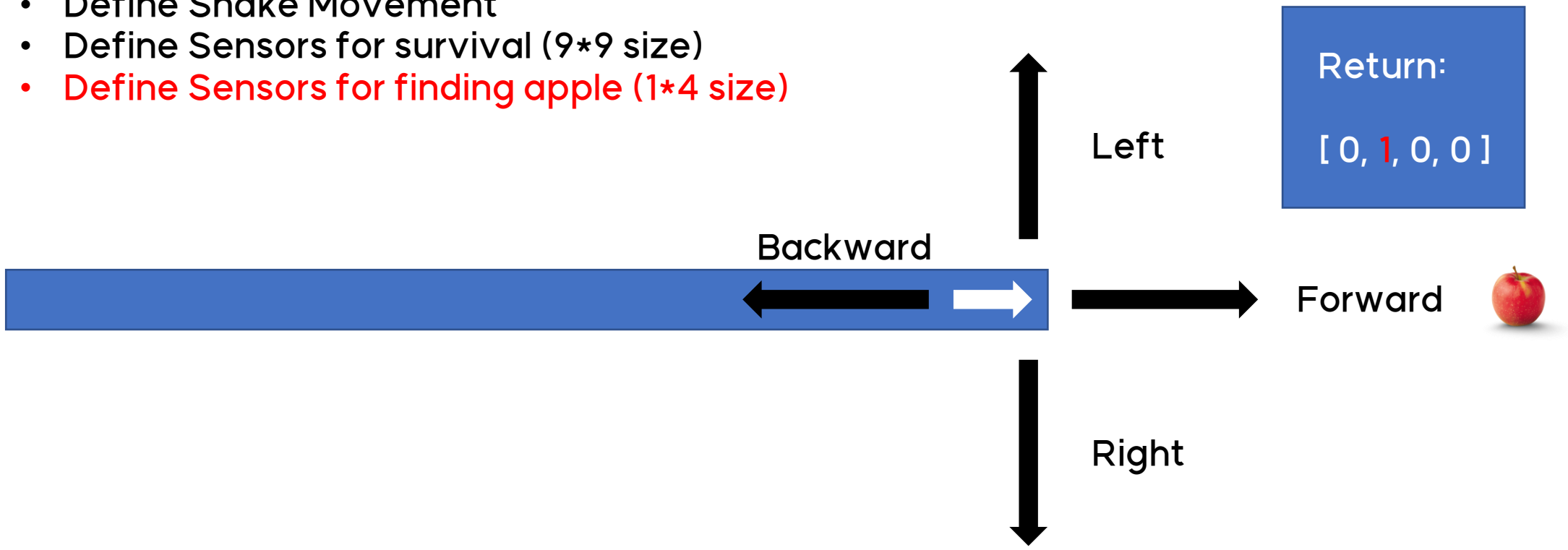
- Define Snake Movement
- Define Sensors for survival (9*9 size)
- Define Sensors for finding apple (1*4 size)



Snake Game with Reinforce learning

Define Sensors Architecture

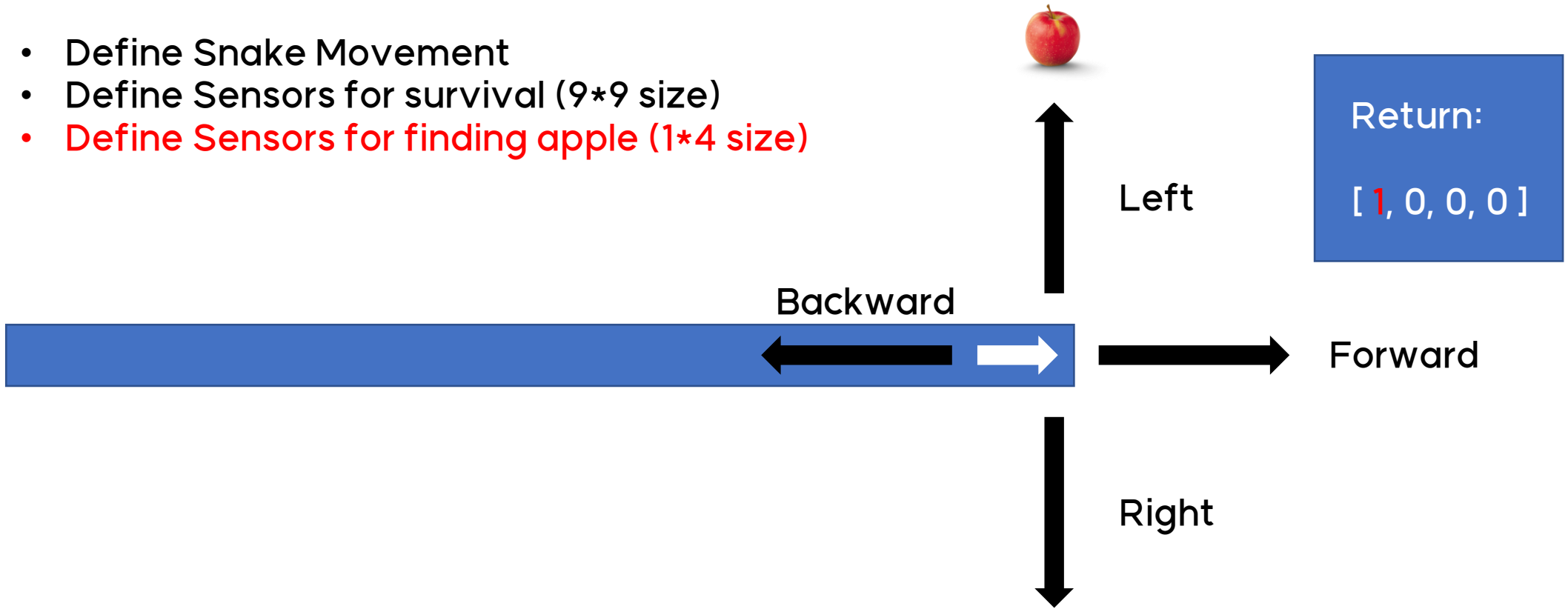
- Define Snake Movement
- Define Sensors for survival (9*9 size)
- Define Sensors for finding apple (1*4 size)



Snake Game with Reinforce learning

Define Sensors Architecture

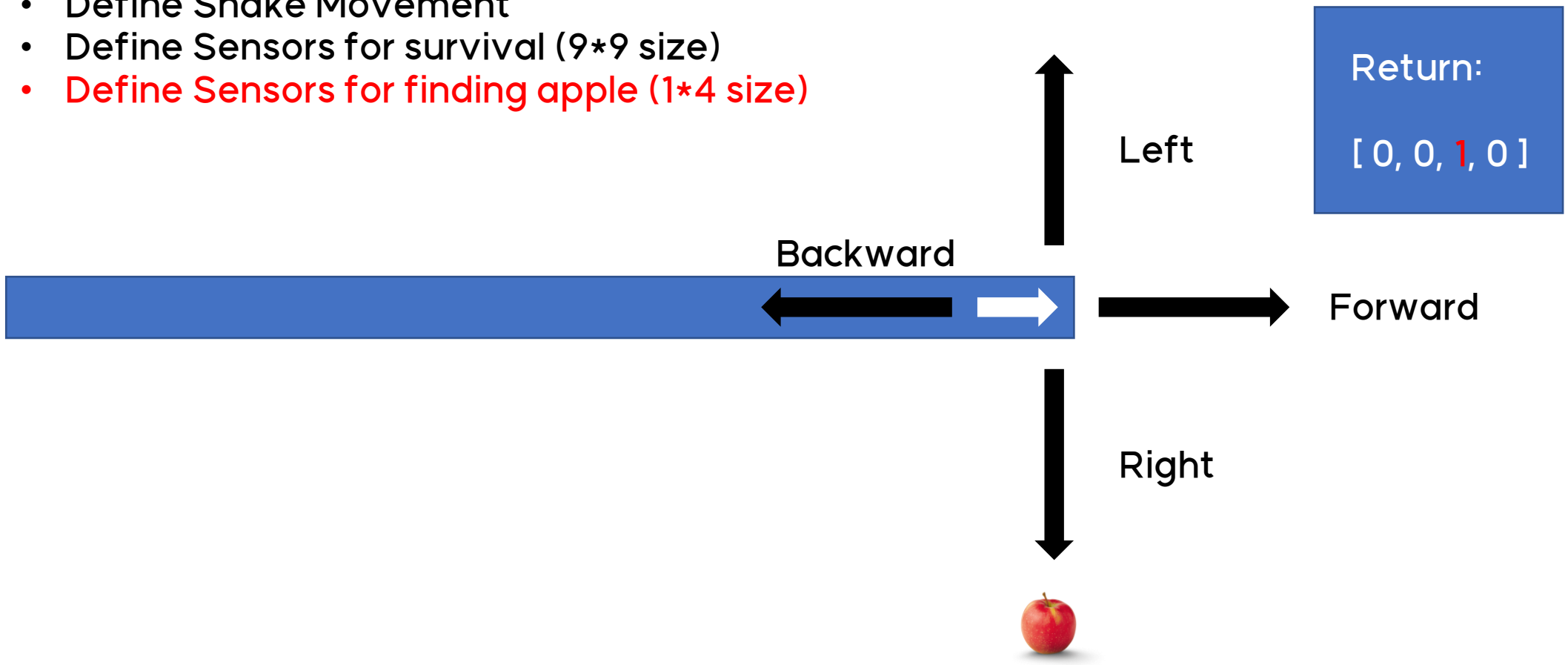
- Define Snake Movement
- Define Sensors for survival (9*9 size)
- Define Sensors for finding apple (1*4 size)



Snake Game with Reinforce learning

Define Sensors Architecture

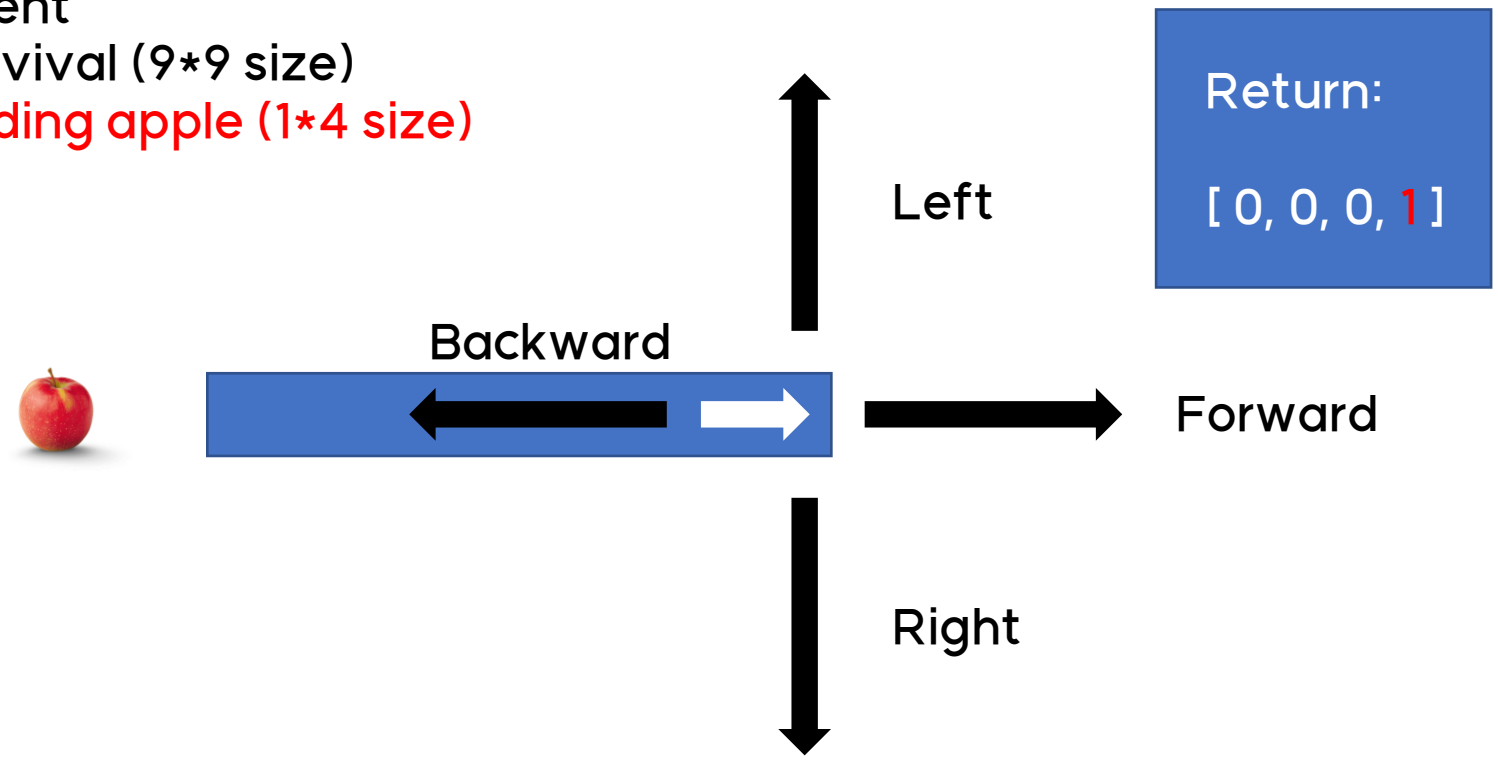
- Define Snake Movement
- Define Sensors for survival (9*9 size)
- Define Sensors for finding apple (1*4 size)



Snake Game with Reinforce learning

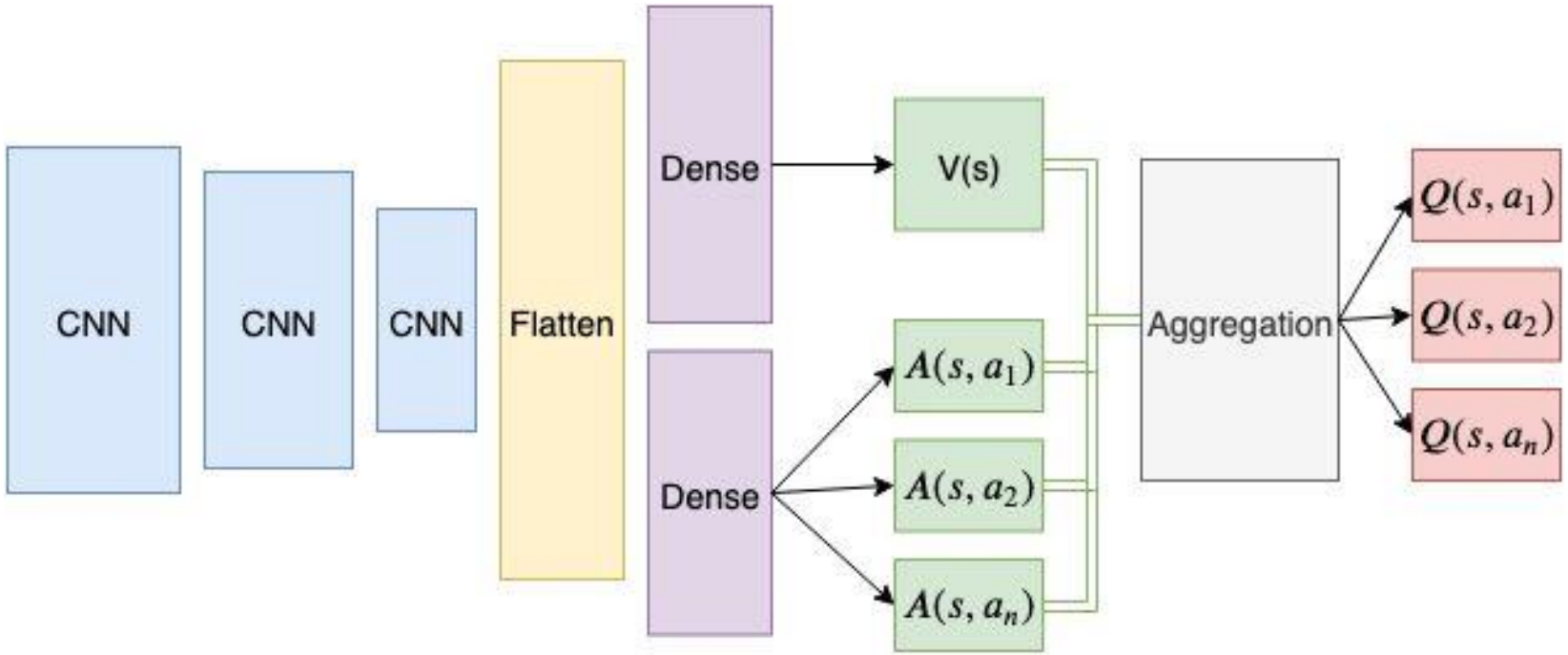
Define Sensors Architecture

- Define Snake Movement
- Define Sensors for survival (9*9 size)
- Define Sensors for finding apple (1*4 size)



Snake Game with Reinforce learning

| Define Layers Architecture



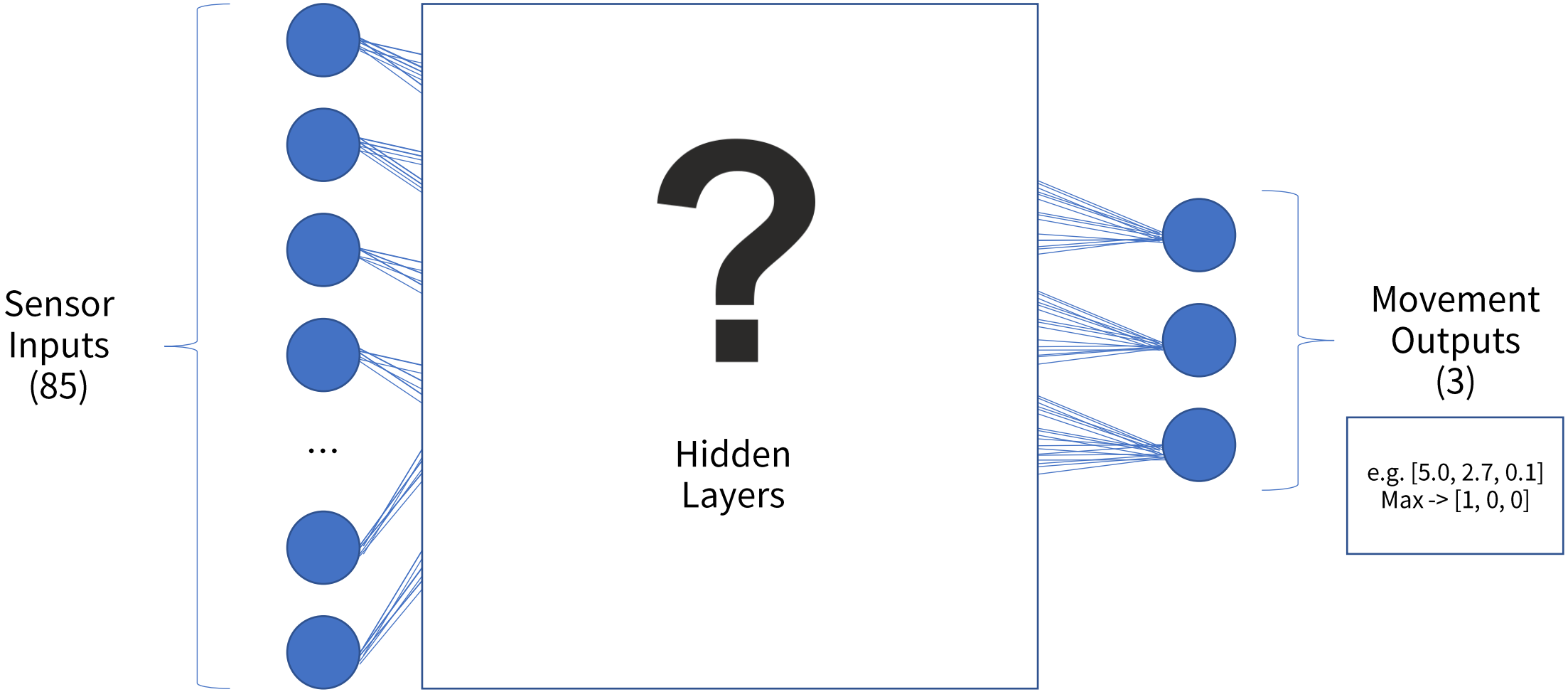
Define Layers Architecture

게임 Board를 입력으로 받지 않고, Sensor를 입력으로 받기 때문에 CNN을 사용하지 않아도 된다.

```
(features): Sequential(
    (0): Linear(in_features=85, out_features=256)
    (1): ReLU(inplace)
    (2): Linear(in_features=256, out_features=128)
    (3): ReLU(inplace)
    (4): Linear(in_features=128, out_features=64)
    (5): ReLU(inplace)
    (6): Linear(in_features=64, out_features=3)
  )
```

Snake Game with Reinforce learning

| Define Layers Architecture



Snake Game with Reinforce learning

Fitness

Reward

1. 먹이를 먹는다. (50 * `body.__len__()`)
2. 먹이와 가까워진다. (5)

Penalty

1. 벽 혹은 자신의 몸에 닿아 죽는다. (-1000)
2. 먹이와 멀어진다. (-1)
3. 위에 조건에 해당하지 않는 행동을 했다. (-0.777)

Snake Game with Reinforce learning

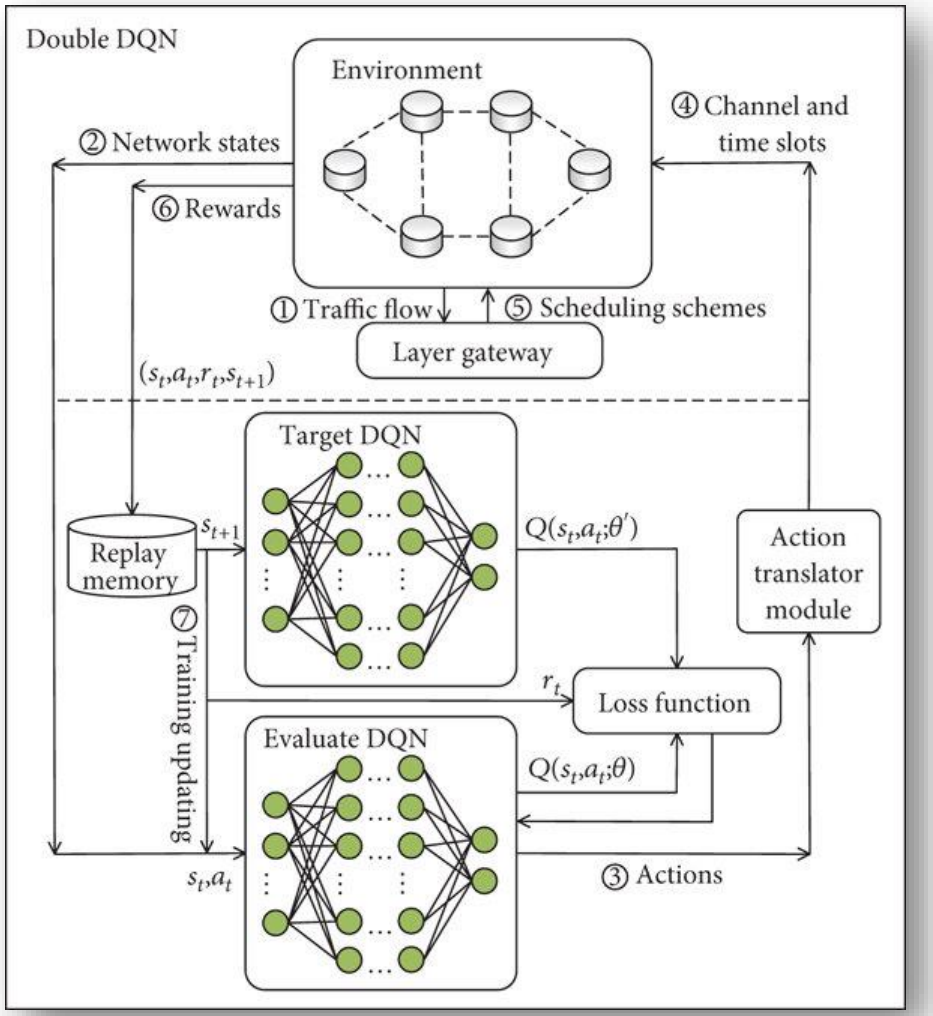
| Limitations

1. 먹이를 향한 의지 vs 살아남기 (자기 몸에 갇히는 문제)
   - 먹이를 향한 의지가 너무 강해서, 몸 안에 있는 사과를 먹기 위해 몸 안으로 들어가는 문제
   - Loophole

2. 오래 걸리는 학습 시간
   - 시간이 지나면 지날수록 오래 걸림 → why? → Model이 학습하면서 생존 시간이 늘어나기 때문.
   - 너무 적은 Episode 횟수라면, 데이터가 잘 모이지 않아, 학습이 잘 되지 않는 문제가 생김.
   - Episode 횟수를 1000회로 잡고 학습을 돌렸을 때, 학습시간이 6시간이 넘어감.

| Concluding…

1. Action 예측 알고리즘을 UCB로 교체
2. Model을 DDQN으로 교체
3. 이전 페이지의 한계점 최대한 줄여보기



Reference

[강화학습 핵심 개념 정리 (1) - wwiiiii 블로그]
- wwiiiii 블로그

[Q-Learning이란? - MangKyu's Diary]
- MangKyu's Diary 블로그

[Deep Q Network (DQN) :: Engineer-Ladder]
- Engineer-Ladder 블로그

[모두를 위한 강화학습]
- 홍콩 과학 기술 대학교 김성훈 교수님의 유튜브 동영상

[Python - 강화학습 Q-Learning 기초 실습 :: Deep Play]
- Deep Play 블로그

[혁펜하임이 들려주는 강화학습]
- 카이스트 전자공학부 혁펜하임 유튜브 동영상

[Playing Atari with deep reinforcement learning.]
- Deepmind 논문

[[Ch.9] DQN(Deep Q-Networks)]
- 숨니의 무작정 따라하기 블로그

[강화학습 논문 정리 3편 : DDQN 논문 리뷰 (Deep Reinforcement Learning with Double Q-learning)]
- 당황했습니까 휴먼? 블로그

[[강화학습 Key Paper] Double DQN]
- Say Hello to the Future블로그

[UCB1 알고리즘의 pseudo-regret 분석]
- choyi0521 블로그