

# Open AI Gym

CartPole 예제로 알아보기

# The CartPole session

## environment 생성

```
import gym
e = gym.make('CartPole-v0')
obs = e.reset()
```

- 새롭게 만든 environment 는 우선 reset 해줘야 함.  
reset() 의 return 값은 첫번째 observation 이다.
- numpy.ndarray 형태로 반환된다.

# The CartPole session

## observation space, action space 확인

```
e.action_space  
e.observation_space
```

- cartPole environment 에서 `action_space` 는 `Discrete(2)`,  
`observation_space` 는 `Box(4, )`이다.  
(각각 막대 무게중심의 x 좌표, 속도, 바닥과의 각도, 각속도)

# The CartPole session

## action 실행

```
e.step(0)
```

0이라는 action 을 취함.

```
e.step(e.action_space.sample())
```

random action을 취함.

- `step()` 은 다음 observation, reward, episode 가 끝났는지에 대한 flag, 기타 정보를 파이썬 튜플 형태로 반환한다.

## sample code

```
import gym

if __name__ == "__main__":
    env = gym.make("CartPole-v0")
    total_reward = 0.0
    total_steps = 0
    obs = env.reset()

    while True:
        action = env.action_space.sample() #
        obs, reward, done, _ = env.step(action)
        total_reward += reward
        total_steps += 1

        if done:
            break

    print("Episode done in %d steps, total reward %.2f"
```

## 실행결과

```
> python3 chap2_cartpole.py  
Episode done in 14 steps, total reward 14.00
```

- agent 가 random action 을 취하기 때문에 값은 매번 다르게 나온다.
- reward boundary : 문제 해결을 위해서 agent 가 받아야하는 reward (100 episodes 의 평균치)
- 이 문제는 reward boundary 가 195인데 위의 코드는 14가 나왔으므로 poor performance 임을 알 수 있다.

# Wrappers

- ObservationWrapper
- RewardWrapper
- ActionWrapper

# Wrappers

## wrapper class 정의

```
class RandomActionWrapper(gym.ActionWrapper):  
    def __init__(self, env, epsilon = 0.1):  
        super(RandomActionWrapper, self).__init__(env)  
        self.epsilon = epsilon
```

- step 에 action을 주었을때, 10퍼센트의 확률로 주어진 action 이 아닌 random action 을 하도록 wrapping 하는 예제.
- `__init__()` 을 상속받아 재정의한다.
- epsilon 은 random action 발생 확률



# Wrappers

## action 함수 정의

```
def action(self, action):  
  
    # 0.1의 확률로 입력 action 대신 random action 발생  
    if random.random() < self.epsilon:  
        print("Random!")  
        return self.env.action_space.sample()  
    return action
```

- `action_space.sample()` 을 사용하여 epsilon 확률만큼 random action 수행.
- 나머지 경우는 입력받은 action 수행.

## main 함수

```
if __name__ == "__main__":  
    env = RandomActionWrapper(gym.make("CartPole-v0"))  
    obs = env.reset()  
    total_reward = 0.0  
  
    while True:  
        obs, reward, done, _ = env.step(0)  
        total_reward += reward  
        if done:  
            break  
  
    print("Reward got: %.2f" % total_reward)
```

## 실행 결과

```
› python3 chap2_cartpole_wrapper.py  
Random!  
Random!  
Random!  
Reward got: 9.00
```

# Monitor

- episode 기록해서 저장할 수 있게 하는 wrapper class.