**강화 학습**

기본 구조 : 에이전트(agent)와 환경(environment)

**에이전트**는 현재 **상태(state)** st를 입력받아 **행동(action)** at를 출력하고 **환경**은 그 행동에 영향을 받아 다음 상태 st+1로 변화한다.

에이전트의 모든 가능한 행동의 집합을 **행동공간(action space)** A이라고 한다. 즉, a∈A이다. 행동공간이 이산적이면 **이산 행동공간(discrete action space)**, 연속적이면 **연속 행동공간(continuous action space)** 이다. 테트리스 게임에서 좌우 이동버튼을 이산 행동공간이라고 할 수 있으며 자동차 운전시 조향핸들각과 페달변위를 연속 행동공간이라고 할 수 있다. 주어진 st에 대하여 at를 결정하는 에이전트의 행동방식을 **정책(policy)** 이라고 한다.

결정론적 정책은 주어진 상태 st에 대하여 결정된 행동 at=μ(st)를 출력한다.

확률론적 정책에서는 행동 at가 확률변수(random variable)이며 확률분포(probability distribution) π(at|st)=Pr(At=at|St=st)를 따른다.[2](https://wikidocs.net/251341#fn:2)[3](https://wikidocs.net/251341#fn:3) 따라서 확률분포로부터 하나의 행동을 결정하기 위한 과정[4](https://wikidocs.net/251341#fn:4)이 추가적으로 필요하다.

 환경의 모든 가능한 상태의 집합을 상태공간 S라고 한다. 즉, s∈S이다. 환경의 상태가 변하는 방식을 모델링한 것을 환경모델(environment model) 또는 **모델(model)** P라고 한다.

정책과 마찬가지로 모델도 **결정론적 모델(deterministic model)** 과  **확률론적 모델(stochastic model)** 로 구분할 수 있다. 결정론적 모델은 현재 상태 st와 행동 at에 따라 다음 상태 st+1이 하나로 결정되지만 확률론적 모델은 st+1이 확률변수이며 확률분포 P(st+1|st,at)를 따른다.

강화학습에서 환경은 주로 **이산시간 MDP(discrete-time Markov Decision Process)** 로 모델링한다. 이산시간 MDP는 이산시간 마로코프 연쇄(discrete-time Markov chain)에 **보상(reward)** 과 선택(decision)을 추가한 모델로써 다음과 같이 순차적 결정 프로세스(sequential decision process) 를 모델링하는 데에 사용된다.

**MDP를 푼다는 것**은 주어진 MDP에 대한 최적정책을 찾는 것을 의미한다. 결국 강화학습은 다음과 같은 최적화 문제라고 볼 수 있다.

1. MDP에 순차적인 행동을 인가하고 순차적인 상태를 획득하여 경로를 생성한다. (이와 같이 획득한 경로를 **샘플(sample)** 이라고 한다.)
2. 샘플에 대한 이득을 계산한다.
3. 1-2 과정을 반복한다.
4. 평균적인 이득을 높일 수 있는 행동방식(정책)을 찾는다.

위는 강화학습에서 알아야 할 최소한의 간단한 이론

**강화학습**

**초기 상태 – 정책을 통한 행동 결정 – 상태 변화**

우리가 만든 모델을 어떻게 강화학습에 사용할까? Cnn-transformer == cnn-lstm(이하 예측모델)

**1) 에이전트의 초기 값 제공**

* CNN-Transformer 역할:
  + 에이전트의 P, I, D 값을 추정.
  + 강화 학습 에이전트가 이를 기반으로 조정.
* 효과:
  + 학습 초기의 랜덤한 행동을 줄이고 학습 속도를 높일 수 있음.
* 이렇게 할거면 예측 모델이 출력하는 pid값을 행동 결정용으로 쓰는 거니까 리워드를 주기 위한 목표 상태를 따로 설정해 줘야 할 듯 ( ex 속도 50km, 직진)

#### 2) ****보상 계산****

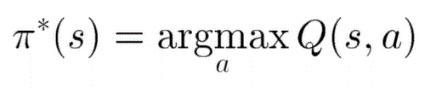
* **CNN-Transformer 역할**:
  + 목표 상태(Target)를 예측하여 보상 계산에 활용.
* **보상 함수 설계 예시**: Reward=−|Predicted\_Target−Current\_State|
* 이 방법으로 할거면 센서값 입력-> 예측모델은 이상적인 pid값 예측 -> 강화학습 모델은 아무 pid나 출력 -> 이 두 pid값을 비교해서 리워드를 주기

로 해야할 듯

가장 많이 활용되는 강화학습 알고리즘: Q-러닝, 사르사(SARSA), 딥 Q 네트워크(DGN)

Q-러닝: 주어진 상태에서 주어진 행동을 수행하는 것이 가져다 줄 효용의 기대값을 예측하는 함수 (최적의 정책을 학습한다)

Q에 내가 가진 상태(state)와 행동(action)을 주면 Q는 이런 상태에서 이런 행동을 했을 때 얼마만큼의 보상을 받을 수 있는 지 알려준다. 그러면 에이전트는 최대 값을 주는 행동을 찾아 그 행동을 하면 된다.

<< Q-러닝의 수식.

\*정책은 보통 파이로 표현한다.

argmax: Q(s, a)를 최대로 하는 action을 찾는다는 뜻(argmax밑의 a가 action을 의미)

파이 옆에 붙은 \*=는 최적의 값임을 뜻함. 즉, Q를 최대로 하는 action을 취하는 정책이 최적임을 의미한다.