RL 스터디

7회차

Sparse 한 보상에서 탐색의 중요성

- 실제 대부분의 환경은 Sparse 한 보상에서 작동된다.

- 일부 아타리 게임 또는 강화학습 환경에서는 연속적으로 또는 몇십 스텝안에 보상을 주는 환경이 많다.
- 하지만 대부분의 경우에는 보상이 매우 Sparse하다.

최소한 보상에 도달 해야 학습이 된다.

- 보상이 랜덤적인 행동으로 도저히 도달할 수 없으면?
- 학습이 계속 안되는데?

사람은 어떻게 학습하는가?

- 어떤 어려운 문제에 직면하면 한번 해봤던 행동, 실패했던 행동은 하지 않는 것이 좋다.
- 사람은 호기심을 가지고 해보지 않은 일을 계속 시도하며 배운다.
- 이것을 내적보상이라고 볼 수 있다.

- 그렇다면 강화학습에도 호기심과 같은 내적보상을 추가한다면 어떨까?

호기심 기반 탐색

- 핵심 아이디어는 미래 예측을 하고 예측오차가 큰 경우에는 그 미래를 안 가봤다고 생각
- 예측 오차 ~~ 내적 보상

미래 예측?

- 픽셀 수준에서의 미래 예측? 관측 수준에서 미래를 예측하는 것은 문제가 존재

- 1. 뉴럴 넷의 특징중 하나로 확률적인 요소가 들어가게 되면 수렴하지 않음
- 2. 실제 행동과는 상관없는 매우 복잡한 모델을 구현해야 될 수 있음

1의 경우

- 어떤 방에 TV가 틀어져 있다고 가정해 보자.
- 랜덤으로 틀어지는 TV의 다음 픽셀을 예측할 수 있는가? => 불가능

- 매우 큰 예측 오차로 다가옴 => 그 자리에서 TV만 보는게 가장 보상이 높은 행동이 된다.

2의 경우

- 나무 옆에 서 있는 사람을 생각하자
- 나뭇잎이 바람에 흔들림 <= 유체 동역학적 모델을 예측 해야함 ???
- 근사가 되니?

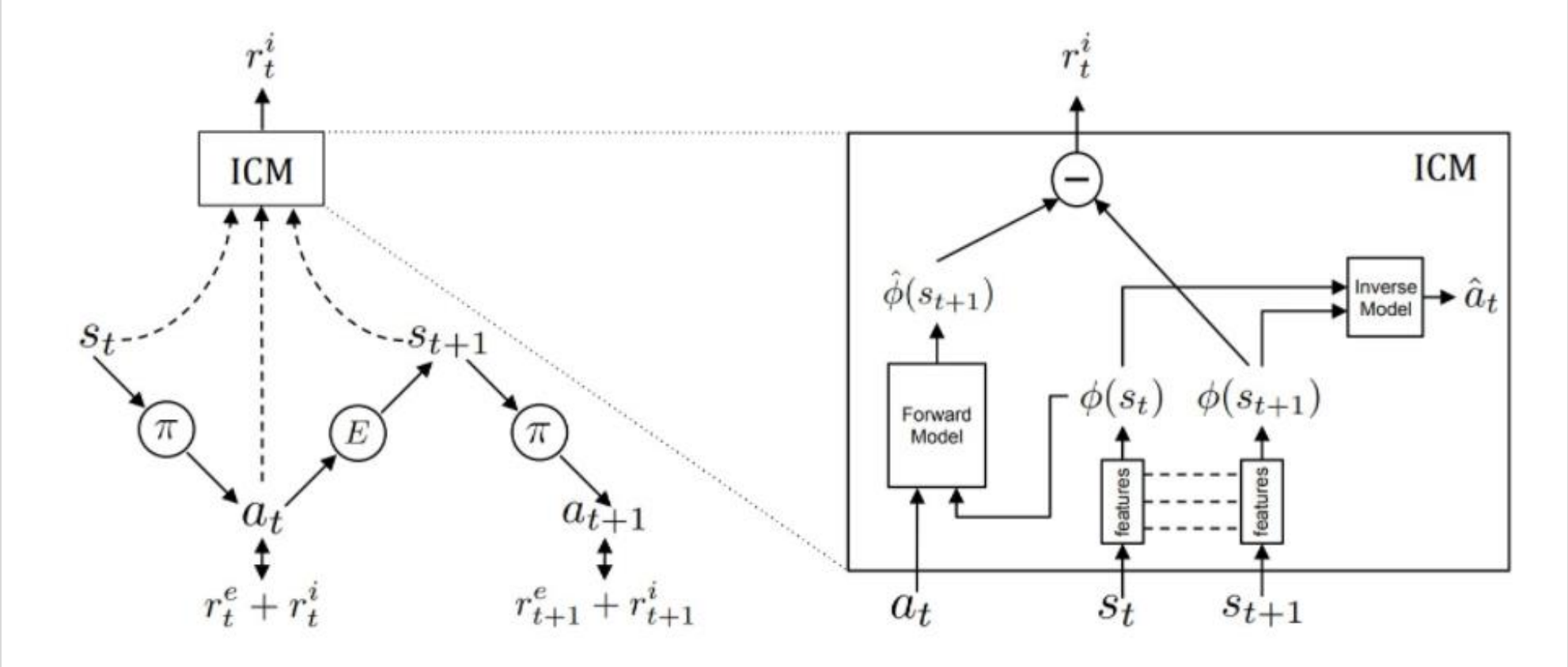
- 뉴럴넷의 정보 수용량을 오버한다 한마디로 이를 근사할 수는 없다. 매우 큰 예측 오차로 다가옴
- 움직이지 않는게 최상의 보상을 받는 행동이 된다.

그러면 어떤 걸 예측한다는 거야?

- 논문에서는 환경요소를 3가지로 나누어 설명했다.
- 1. 에이전트가 제어 가능한 요소
- 2. 에이전트가 제어 불가능하지만 행동에 영향을 주는 요소
- 3. 에이전트가 제어 불가능하고 행동에도 영향을 주지 않는 요소

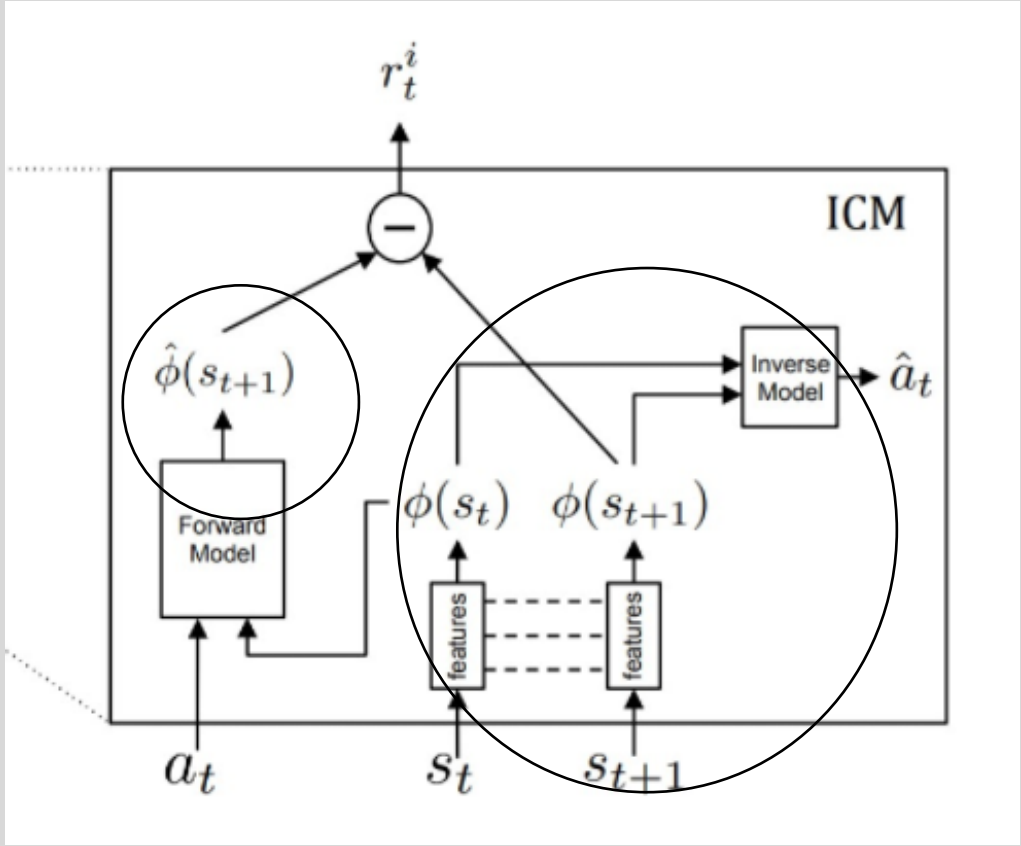
- 1번 2번의 특징적 요소를 가진 새로운 특징 공간 내에서 미래를 예측하면 된다.
- 이러한 특징 공간을 근사하는 시스템과 그리고 예측 오차를 나타내는 시스템을 통합

호기심 기반 탐색


r_t^i
ICM
s_t
s_{t+1}
π
E
π
a_t
a_{t+1}
$r_t^e + r_t^i$
$r_{t+1}^e + r_{t+1}^i$
r_t^i
ICM
$\hat{\phi}(s_{t+1})$
Inverse Model
$\hat{a}_t$
Forward Model
$\phi(s_t)$
$\phi(s_{t+1})$
features
features
a_t
s_t
s_{t+1}

호기심 기반 탐색

- 상태를 넣으면 특징 벡터를 리턴하는 특징네트워크
- 특징 벡터를 입력으로 받아서 행동을 출력하도록 하는 inverse model을 학습 => 그라디언트 흐름으로 행동과 관련된 특징 벡터가 추출 될 수 있도록 학습이 됨
- 그리고 Forward Model을 통해 특징 벡터와 행동을 입력 받고 예측 그리고
- 예측 값의 MSE 값이 호기심 보상이 된다.



호기심 기반 탐색

- 이렇게 얻은 내적 보상을 외적보상과 적절한 비로 더하여 보상으로 사용

random network distillation

- 호기심 기반 탐색 방식에는 문제가 있다 => 예측을 한다는 것 자체가 불 안정 + inverse model 학습도 불안정
- 성능이 조끔 튄다.

random network distillation

- 호기심 기반 탐색 방식에는 문제가 있다 => 예측을 한다는 것 자체가 불 안정 + inverse model 학습도 불안정
- 성능이 조끔 튄다.

- 예측 자체를 하지 말자

random network distillation

- 어떠한 비슷한 상태를 많이 학습시킬수록 오차가 낮아진다.

- 여기서는 예측 오차에 기여하는 주요 특징을 4가지로 분류했다.

- 1. Amout of training data – 유사한 상태의 학습 데이터의 양
- 2. Stochasticity – 목표함수의 확률적 전이 요소
- 3. Model misspecification – 모델 불일치
- 4. Learning dynamics – 근사 학습 자체가 실패

- 이중 2번과 3번의 요소를 RND는 제거한다.

random network distillation

- 동일한 아키텍쳐를 가진 랜덤하게 초기화된 타겟 네트워크
- 그리고 계속 학습시킬 예측 네트워크 2가지를 만들어 낸다.

=> 예측 네트워크는 타겟 네트워크와 같은 출력을 내도록 학습되고 두 네트워크의 오차 만큼이 내적 보상이 된다.

random network distillation - 정규화

하지만 이러한 보상의 스케일이 다르기에 일반화가 힘들다 => 예측 오류를 정규화 한다.

신경망의 입력 값 또한 정규화를 하고 clip 하여 과도한 입출력의 변화를 막는다.

random network distillation

Algorithm 1 RND pseudo-code

$N \leftarrow$ number of rollouts
$N_{\text{opt}} \leftarrow$ number of optimization steps
$K \leftarrow$ length of rollout
$M \leftarrow$ number of initial steps for initializing observation normalization
$t = 0$
Sample state $s_0 \sim p_0(s_0)$
for $m = 1$ **to** M **do**
  sample $a_t \sim \text{Uniform}(a_t)$
  sample $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$
  Update observation normalization parameters using s_{t+1}
  t += 1
end for
for $i = 1$ **to** N **do**
  for $j = 1$ **to** K **do**
    sample $a_t \sim \pi(a_t|s_t)$
    sample $s_{t+1}, e_t \sim p(s_{t+1}, e_t|s_t, a_t)$
    calculate intrinsic reward $i_t = \|\hat{f}(s_{t+1}) - f(s_{t+1})\|^2$
    add $s_t, s_{t+1}, a_t, e_t, i_t$ to optimization batch B_i
    Update reward normalization parameters using i_t
    t += 1
  end for
  Normalize the intrinsic rewards contained in B_i
  Calculate returns $R_{I,i}$ and advantages $A_{I,i}$ for intrinsic reward
  Calculate returns $R_{E,i}$ and advantages $A_{E,i}$ for extrinsic reward
  Calculate combined advantages $A_i = A_{I,i} + A_{E,i}$
  Update observation normalization parameters using B_i
  for $j = 1$ **to** N_{opt} **do**
    optimize θ_π wrt PPO loss on batch B_i, R_i, A_i using Adam
    optimize $\theta_{\hat{f}}$ wrt distillation loss on B_i using Adam
  end for
end for

끝