**A2C 알고리즘**

AC에서는 두 개의 신경망을 훈련합니다.

* REINFORCE 알고리즘과 유사한 정책 네트워크
* DQN과 유사한 상태-가치 네트워크

따라서 Policy Network가 주요 영웅 역할을 하고 State-Value Network가 비평가 역할을 하는 Actor-Critic이라는 이름이 붙었습니다. GAN에 대해 읽어본 적이 있다면 이 개념이 적대적 시스템에 생성자와 판별자가 포함되어 있는 경우 다소 친숙하게 들릴 수 있습니다.

Policy Gradient 알고리즘의 손실 함수는 다음에서 업데이트됩니다.

*손실 = -1 x Σlog(확률) x 할인\_보상*

*에게*

*손실 = -1 x Σlog(확률) x (보상+****γ****V(S`)-V(S))*

그리고 DQN의 경우 평균 제곱 오류로 유지됩니다. 따라서 최종 손실은 다음과 같습니다.

*-1 x Σlog(확률)x(보상+****γ****V(S`)-V(S))+MSE(실제\_V(S), 예측\_V(S))*

여기서 용어 *(Reward+****γ****V(S`)-V(S))는***Advantage** 용어 라고 불리는 State-Value Network에서 유래하므로 Advantage Actor-Critic이라는 이름이 붙습니다. 자세히 살펴보면 이는 우리가 DQN과 Q-Learning에서 사용한 Bellman 방정식에서 나온 것입니다. 그래서 우리가 실제로 하고 있는 일은

* DQN 및 REINFORCE 신경망 모두에서 예측합니다.
* 두 네트워크에 대한 조건이 포함되므로 새로운 손실을 계산합니다.
* 가중치 업데이트를 위해 두 신경망 모두에 역전파합니다.

코드를 시작하기 전에 알아야 할 개념이 하나 더 있습니다.

**N단계 학습**

N Step 학습에 뛰어들기 전에, 우리는 무엇이 무엇인지 알아야 합니다.

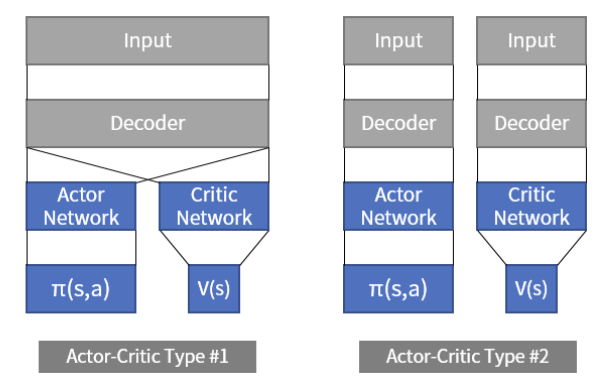
**온라인 학습:** 모든 조치가 취해진 후 에이전트 업데이트(경험 재생이 없는 DQN)

**Monte Carlo:** 에피소드 종료 후 에이전트 업데이트(REINFORCE)

N단계 학습은 에이전트가 'x' 단계를 수행할 때 에이전트를 업데이트하는 두 단계 사이의 것으로, 여기서 'x'는 Actor-Critic 방법에 적용할 설정할 수 있는 임계값입니다.

Actor는 상태가 주어졌을 때 행동을 결정하고, Critic은 상태의 가치를 평가합니다

Actor-Critic 은 마지막에 값을 합치지 않는다

**

step 마다 얻어진 상태(s), 행동(a), 보상(r), 다음 상태(s’)를 이용해서 모델을 학습

Actor-Critic 은 π(s,a)값과 V(s)값을 구합니다. V(s)는 지금까지 계속 다뤘던 가치함수이고, π(s,a)는 어떤 상태에서 특정 행동을 취할 확률

policy 는 에이전트가 어떤 행동을 취할지에 대한 정책이라는 뜻이고, gradient 는 미분을 통해 policy 값을 업데이트하며 최적의 policy 를 찾아간다는 의미입니다. 그런데 이렇게 에이전트의 행동 확률을 직접적으로 학습하는 방법은 불안정하기 때문에 가치함수를 같이 써서 안정성을 높이는 것이 Actor-Critic 의 핵심

Actor-Critic 의 Actor 의 기대출력으로 Advantage 를 사용하면 Advantage Actor-Critic, A2C 가 됩니다. Advantage 는 예상했던 것(V(s)

)보다 얼마나 더 좋은 값인지를 판단하는 값으로, Q(s,a)

에서 V(s)

를 빼준 값을 많이 사용합니다.

A(s,a)=Q(s,a)−V(s)

그런데 Q(s,a)

를 구하는 부분은 그림 1의 Actor-Critic 알고리즘 구조에 나와있지 않습니다. 하지만 이 시리즈의 두번째 글에서 다뤘던 내용처럼,

실제가치≃현재가치+미래가치

이고 현재 가치는 보상, 미래 가치는 다음 상태의 가치 함수 V(s′)

로 치환한다면 아래와 같은 식을 얻을 수 있습니다.

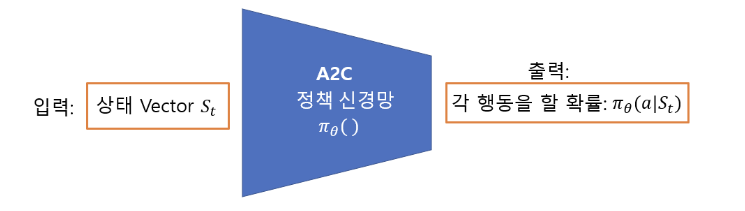
Q(s,a)≃R+γV(s′)

정리하면 Advantage 를 구하는 식은 아래와 같이 바뀝니다.

A(s,a)≃R+γV(s′)−V(s)

Critic Network 에서 계산한 V(s)

값이 Actor Network 의 계산에도 영향을 끼치게 됩니다.



π(s,a)는 어떤 상태에서 특정 행동을 취할 확률입니다. 보통 이런 확률은 softmax 를 사용해서 얻어낼 수 있습니다. Softmax 는 일정한 값들을 e를 밑으로 하는 지수로 계산한 다음 합치고 나눠서 합이 1.0 인 확률로 변환시켜줍니다.

Actor 의 기대출력으로 Advantage 를 사용하면 Advantage Actor-Critic, A2C 가 됩니다. Advantage 는 예상했던 것(V(s))보다 얼마나 더 좋은 값인지를 판단하는 값으로, Q(s,a)에서 V(s)를 빼준 값을 많이 사용합니다.

A(s,a)=Q(s,a)−V(s)

Q(s,a) 실제가치≃현재가치+미래가치이고 현재 가치는 보상, 미래 가치는 다음 상태의 가치 함수 V(s′)로 치환한다면 아래와 같은 식을 얻을 수 있습니다.

Q(s,a)≃R+γV(s′)

정리하면 Advantage 를 구하는 식은 아래와 같이 바뀝니다.

A(s,a)≃R+γV(s′)−V(s)