강화학습 과제

202201271 김서현

(a)

MachineEnv Class: 단일 설비의 유지보수를 위한 구체적인 문제 환경을 정의해 놓은 코드이다. State와 Action, Reward, State transition 등 최적 정책 수립에 필요한 기계 설비의 환경을 정의해 놓았다.

상태는 부품의 마모 정도와 경과 시간으로 구성된 3차원, Action은 유지보수 결정 여부에 따라 4개, Reward는 제품이 정상적으로 생산되면 +20, 고장나면 -100, 부품 교체 시 -10의 보상이 발생되게끔 설정되어 있다. MachineEnv Class는 이와 같이 생성자 메서드에서 정의된 것을 기반으로 구동된다. 유지 보수하지 않고 설비를 가동했을 땐, Uniform distribution에서 생성된 난수가 failure_prob보다 작을 경우 설비가 고장난 경우로 판단, 난수가 failure_prob보다 큰 경우는 정상적으로 제품이 생산되어 20의 수익이 발생한 것으로 판단한다. 부품 1을 교체하면, 교체비용 10이 발생하고 부품 1의 마모 정도는 0으로 리셋된다. 부품 2를 교체하면, 교체비용 10이 발생하고 부품 2의 마모 정도는 0으로 리셋된다. 부품 1과 부품 2를 모두 교체하면 20의 교체비용이 발생하고 두 부품의 마모 정도는 0으로 리셋된다. 각 step이 발생할 때 마다 시간은 1씩 증가한다. 각 부품의 마모 정도는 0과 1사이 값이며, 상태 3, 즉 경과 시간이 100분이 되면 현재 State와 Return을 리턴하고 종료된다.

DQN Class: 위 Class는 각 State에 대한 Action의 Q값을 예측하기 위한 Deep Neural Network 모델링 코드이다. 3차원의 Input과, 2개의 hidden layer, 4차원의 output으로 이루어져 있다. Activation function은 ReLU를 사용했다. 순전파 연산을 위한 forward메소드가 포함되어있다.

ReplayBuffer Class: DQN에서 ReplayBuffer는 sequential state로 인한 bias발생의 문제점을 개선하고자 제시된 방안이다. 초기화 된 (s,a,r,s′) 저장소에 각 step에서 발생한 (s,a,r,s′)샘플들을 self.count위치에 max_size만큼 저장한다. 에피소드 종료 여부도 함께 저장한다. 이후 샘플들이 저장된 buffer에서 batch_size만큼 랜덤 샘플링을 진행한 후 텐서로 변환한다.

DQN_Learning Class: Deep Q – network Learning을 진행한다. 생성자 메소드에서 Q값 예측을 위한 q와 target network Q인 q_target을 정의한다. Optimizer와 loss함수, 하이퍼파라미터도 정의한다. Step메소드에서 (s,a,r,s′), done데이터를 ReplayBuffer저장한다. ε – $greedy$ 정책으로 1- ε 의 확률로 Q값이 가장 큰 Action을 선택한다. Learn메소드에선 64보다 크거나 같은 batch_sample에 대해 종료 상태가 아닌 경우 target Q값을 계산한

다. Current network의 Q값도 계산한다. MSE 손실을 계산하고, 역전파로 네트워크의 파라미터를 업데이트한다. Target network는 soft_update방법으로 천천히 업데이트 한다. soft_update는 $\theta' = \tau\theta + (1-\tau)\theta'$ 이렇게 이루어진다. 위와 같은 학습을(learn method) K 간격으로 반복한다.

(b)

<Hyperparameter>

num_episode = 300

max_steps = 100

epsilon_start = 0.4

epsilon_end = 0.1

epsilon_decay_rate = 0.9

gamma = 1

K = 32

tau = 0.005

lr = 0.001

freq = 50

Episode 50: Average rewards for the last 50 episodes: 374.60

Episode 100: Average rewards for the last 50 episodes: 423.40

Episode 150: Average rewards for the last 50 episodes: 461.60

Episode 200: Average rewards for the last 50 episodes: 373.80

Episode 250: Average rewards for the last 50 episodes: 217.20

Episode 300: Average rewards for the last 50 episodes: 47.80

(c)

주기적 유지보수 휴리스틱을 사용하였을 때 가장 좋은 유지보수 기간 조합:

Part 1 = 3일, part 2 = 6일

(d)

조건적 유지보수 휴리스틱을 사용하였을 때 가장 좋은 유지보수 임계치 조합:

Part 1 = 0.30, part 2 = 0.35

(e)

100개의 에피소드를 거쳤을 때

DQN 기반 정책 reward: -1021

주기적 유지보수 휴리스틱 reward: -223.1

조건적 유지보수 휴리스틱 reward: -521.6

DQN 기반 정책 reward가 매우 작다. 이는 유지보수를 하지 않아 고장이 매우 빈번하게 일어나고 있음을 추론할 수 있다. 300번의 에피소드를 돌렸을 땐 좋은 성능을 보인 반면 100번의 에피소드에서는 평균 reward가 매우 낮은 것을 보아 학습이 많이 진행될수록 더 좋은 성능을 보이는 것을 확인할 수 있다.

휴리스틱을 적용했을 때는 naive한 DQN 기반 정책보다 성능이 좋다. 주기적, 조건적으로 유지보수를 하도록 함으로써 고장을 미연에 방지할 수 있어 더 좋은 성능을 보일 수 있다. 두 가지 휴리스틱 중에서는 주기적 유지보수 휴리스틱의 성능이 더 좋다. 설비의 마모 속도가 불확실성이 크다면 주기적으로 유지보수를 진행함으로써 고장을 방지할 수 있다.