

Dueling DDQN_MultiStepLearning_main

1. 모듈 임포트, 초기값 세팅(epsilon decay까지 plotting)
2. Class ModifiedTensorBoard(TensorBoard) : 사용안함
3. Class UAVEnv
 - Def reset
 - 에피소드의 스텝 = 0
 - Return random_generate_states
 - Def random_generate_states
 - Return 50~1950 중 랜덤 x,y 좌표 생성
 - Def get_empirical_outage
 - Return ran_env.getPointMiniOutage
 - Def step
 - Step += 1, next_step 정하기, outbound 정하기
 - Terminal 조건 설정, reward 설정, done 설정
 - Return next_step, reward, terminal, outbound, done
4. Seed값 설정
5. Class DQNAgent
 - Def init
 - Model, target model 생성,
 - Initialize model, replay memory 생성
 - Def create_model
 - Layer 설정(with Deulilng) + compile + summary
 - Def normalize_data
 - Def initialize_model : DQN 초기화 so as to Qvalue equals to 논문에 나오는 식
 - next state 생성 based on 100,000 random 좌표 생성, action_aug 생성(1000 x 2 x 4)
 - 이 값을 토대로 Q_init 생성
 - Train, test data(8 : 2비율) -> model.fit -> MSE, MAE plotting,
 - Return model.evaluate(test data)

- Def update_replay_memory_nStepLearning(self, slide_window, nSTEP, endEpisode)
 - Update only after n steps
 - Slide_window = current_state + action_idx + reward + next_state + terminal + outbound + done
 - Slide_wind에서 위의 7개 받은 후 replaymemory.append
 - 에피소드 끝의 마지막 몇 단계에 대한 잘린 n-step return
- Def sample_batch_from_replay_memory : replaymemory에서 minibatch sampling
- Def deepdoubleQlearn
 - Start training only if certain number of samples is already saved
 - Replay memory sample한 다음 current, next Q value, next action 설정
 - For in minibatch_size : target value 업데이트 -> model.fit
 - 에피소드 끝날 때마다 counter += 1, counter >= 5면 target network weight update
- Def choose_action(current_state, cur_traj, epsilon)
 - Next_possible_state 생성, action은 이미 지나온 state(=trajectory) 다시 가는 것 방지
 - Epsilon decay에 따라 action 선택

6. 에피소드 진행

- Current state 기준 trajectory, action 생성 -> env.step
- While not done : Reward, slide_window, replay memory update, deedoubleqlearn

7. Moving average : 정의 및 plotting

8. 결과값 plotting

radio_environment

1. 모듈 임포트, 초기값 세팅
2. 빌딩 distribution : 모든 빌딩의 2차원 좌표, height 구함
3. BS distribution
4. Def getAntennaGain : 3차원에서의 antenna gain 구현(참고논문 [3]. [5])
5. Def getPointMiniOutage : Main Function that determines the best outage from all BS at a given location
6. Def getPointOutageMatrix : 해당좌표에서 모든 BS의 모든 sector에 대한 empirical outage 확률 리턴
7. Def getReceivePower_RicianAndRayleighFastFading : 1개 BS의 3개 sector로부터의 received power 리턴

8. Def checkLoS : 해당 좌표와 BS가 LoS가 있는지?
9. Def getLargeScalePowerFromBS : Large scale path loss power
10. View the radio map for given height