프로젝트 개요

- 목표
  - RLlib을 활용해 MuJoCo Continuous Control 환경(HalfCheetah-v5)에서 강화학습(PPO)의 성능, 효율성, 운영 안정성을 실험적으로 분석
- 핵심 실습 요소
  - 환경 병렬화가 학습 효율성에 미치는 영향
  - 하이퍼파라미터 변화에 따른 수렴 속도·안정성 비교
  - 학습된 정책을 활용한 **모델 평가 함수 구현**
- 기간
  - rllib_mujoco.py 와 rllib_mujoco_compute_action.py 활용
  - 2주 + α (Due : 11/17(월) 23:59)
- 환경
  - RLlib (Ray v2.50.1)
  - Gymnasium 1.2.1
  - MuJoCo 3.3.7

실험 환경

- 환경
  - HalfCheetah-v5
- 알고리즘
  - PPO (RLlib)
- 공통 설정
  - 로그 항목
    - episode_reward_mean, SPS, time_this_iter_s
  - 리소스 모니터링
    - CPU, GPU, RAM, VRAM
  - 5회 반복 후 평균 ± 표준편차 산출
    - 시드는 본인 학번, 학번 + 1, ..., 학번 + 4 (debugging(seed=xx))
- 가시화:
  - Reward vs. Steps, SPS vs. Steps, Time vs. Iteration 그래프

실험 설계

- (1) 병렬화 효율성 분석
  - 변경 요소
    - num_env_runners
    - num_envs_per_env_runner
  - 측정 항목
    - time_this_iter_s
    - SPS
    - CPU/GPU/RAM/VRAM utilization
  - 분석 포인트
    - 병렬화 수준 증가에 따른 처리량 향상 한계점 및 자원 병목 분석

실험 설계

- (2) 학습 안정성 및 성능 비교
  - 변경 파라미터
    - 우측 파라미터 외 변경
  - 지표
    - 성능 (5회 평균)
    - 안정성 (5회 분산)
  - 분석 포인트
    - 하이퍼파라미터 변화에 따른 학습 성능 및 안정성 비교

```
.training(
    # Following the paper.
    # ray/rllib/tuned_examples/ppo/benchmark_ppo_mujoco.py
    lambda_=0.95,
    lr=0.0003,
    num_epochs=15,
    train_batch_size=32 * 512,
    minibatch_size=4096,
    vf_loss_coeff=0.01,
    model={
        "fcnet_hiddens": [64, 64],
        "fcnet_activation": "tanh",
        "vf_share_layers": False,
    },
)
```

실험 설계

- (3) 모델 평가 함수 구현
  - 목표
    - 학습된 RLlib 정책을 이용해 새로운 에피소드 평가 수행
    - compute_action(obs) 함수 구현
    - RLlib으로 학습된 모델 불러와서 obs 기반 action 반환

```
NUM_EVAL_EPISODES = 10
returns = []

def compute_action(obs):
    action = algo.compute_single_action(obs,
explore=False)
    return action

for ep in range(NUM_EVAL_EPISODES):
    obs, info = env.reset()
    done = False
    ep_ret = 0.0
    while not done:
        action = compute_action(obs)
        obs, reward, terminated, truncated, info =
env.step(action)
        done = terminated or truncated
        ep_ret += float(reward)
    returns.append(ep_ret)
    print(f"[EVAL] Episode {ep+1}/{NUM_EVAL_EPISODES}:
return={ep_ret:.3f}")

print(f"평균 리턴: {sum(returns)/len(returns):.3f}")
```

보고서 구성

- 서론: HalfCheetah-v5 환경 소개(개요,관측,행동,보상 등) + PPO 개요 + 보고서 목적 및 배경 등
- 방법: 환경, 하이퍼파라미터, 실험 설계
- 실험 결과
  - 병렬화 분석, 학습 곡선, 평가 결과 통계
  - 학습 성능 vs. 평가 성능
  - 병목 원인, 성능, 안정성, 성능·안정성 trade-off 분석
- 결론: 학습 효율 향상 및 학습 제언

평가지표

- 제출물(압축본)
  - ray_results 폴더 (온전한 5개만 남길 것)
  - 결과보고서
  - 코드
- 평가지표

항목	배점	내용
(1) 병렬화 효율성 분석	4	실험 다양성 및 결과
(2) 학습 안정성 및 성능 비교	3	실험 다양성 및 결과
(3) 모델 평가 함수 구현	6	동작 여부
보고서 내 분석	6	실험 결과와 분석 간 논리적 일치 분석 깊이 및 다양성
보고서 완성도	6	미적 우수성 (형태, 구성, 표, 그림 등)
총점	25	