Reinforcement Learning for Autonomous Driving in Highway Environments

1. Overview

My project uses reinforcement learning (RL) algorithm to train an autonomous driving agent in the `highway-v0` environment, part of the `highway-env` suite integrated with `Gymnasium`. The objective is to teach the agent how to navigate multi-lane traffic safely and efficiently using the Proximal Policy Optimization (PPO) algorithm.

2. Algorithm: Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is a widely used model-free, on-policy RL algorithm. It is favored for its balance between performance and implementation simplicity. PPO is a policy gradient method that improves stability by using a clipped surrogate objective function to limit the magnitude of policy updates.

Key features:

- **Clipped objective function** to restrict policy updates within a small trust region.
- **Advantage estimation** using Generalized Advantage Estimation (GAE) for lower variance and better learning signals.
- **Multiple epochs of mini-batch updates** to improve sample efficiency.

Hyperparameters used:

- `learning_rate = 1e-3`
- `n_steps = 128`
- `batch_size = 16`
- `n_epochs = 2`
- `gamma = 0.98` (discount factor)
- `gae_lambda = 0.90` (for GAE)
- `clip_range = 0.2` (PPO clipping)
- `ent_coef = 0.01` (entropy regularization for exploration)
- `policy network architecture = [32, 32]` (2-layer MLP with 32 neurons each)

3. Environment Configuration

The project uses multiple environments (`highway-v0` and `merge-v0`) with kinematics-based observations. The environment simulates vehicles navigating on a highway with various parameters:

python
CopyEdit
```
{
  "observation": {
    "type": "Kinematics",
    "vehicles_count": 5,
    "features": ["presence", "x", "y", "vx", "vy"],
    "normalize": True
  },
  "lanes_count": 2 or 3,
  "vehicles_count": 10 or 30,
  "duration": 10,
  "collision_reward": -1,
  "reward_speed_range": [20, 30]
}
```

These configurations allow the agent to learn policies based on positions and velocities of nearby vehicles.

4. Training Setup and Monitoring

Training is managed using `stable-baselines3`'s PPO implementation and custom monitoring:

- The environment is wrapped with a `Monitor` to record episode statistics.
- A custom callback `QuickMetricsCallback` tracks:
    - Episode rewards
    - Episode lengths
    - Crash frequency
- Intermediate metrics are printed every 100 episodes, including crash rate and recent rewards.

Training runs for `5000` timesteps, repeated twice with different environment complexities (e.g., increased lane and vehicle count in the second phase).

5. Evaluation

With original configuration:

Episodes completed: 552
Mean reward: 6.61
Final reward (last 2): 7.09
Crash rate: 21.9%
Mean episode length: 9.3

With custom traffic scenario:

Episodes completed: 519
Mean reward: 6.95
Final reward (last 2): 7.03
Crash rate: 6.4%
Mean episode length: 9.8