CS 778: Assignment 1

~Shriya Garg(221038)

Objective

The goal of this report is to implement and compare three key policy gradient algorithms from the course: Vanilla Policy Gradient (VPG), Trust Region Policy Optimization (TRPO), and Proximal Policy Optimization (PPO). We will test their performance, stability, and learning speed on five different control tasks from the Gymnasium library.

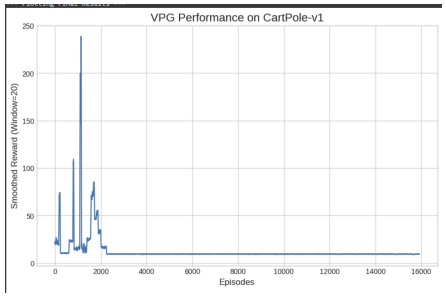
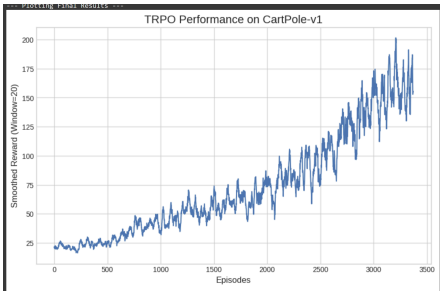
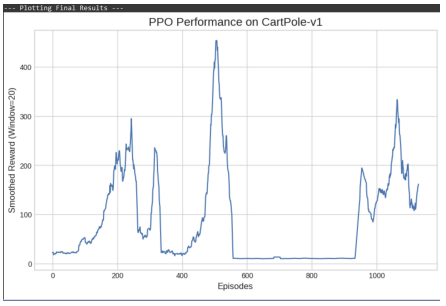
Methodology

All three algorithms were implemented from scratch in PyTorch, following the principles discussed in the lecture notes. A consistent Actor-Critic architecture with Generalized Advantage Estimation (GAE) was used across all implementations to ensure a fair comparison.

Results and Analysis

1. CartPole-v1

This is a simple environment where the goal is to balance a pole on a cart. The performance of the three algorithms is shown below.

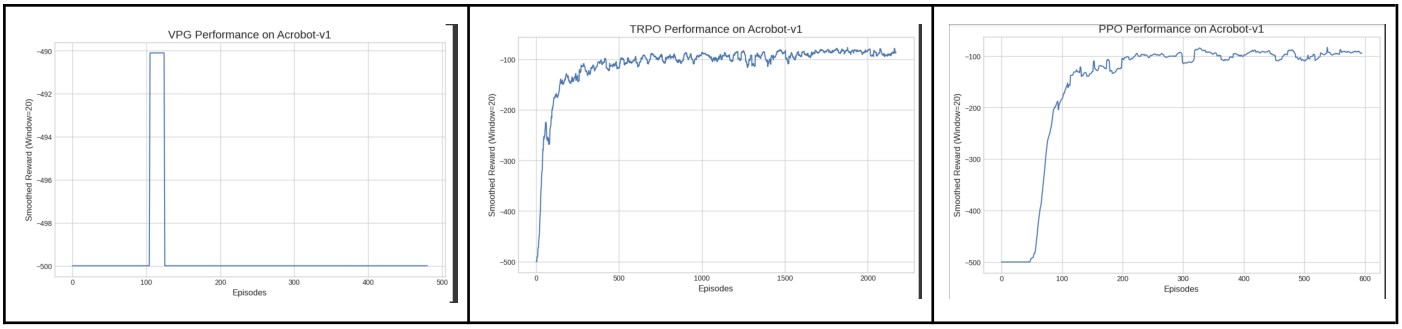
VPG	TRPO	PPO-clip
		

Analysis: On this simple task, TRPO and PPO algorithms eventually learn, but their stability differs dramatically. The VPG graph shows extreme instability; its performance spikes but then collapses, a classic example of the "high variance" updates discussed in the lecture notes. TRPO provides the most stable learning curve, showing a smooth, steady improvement as its trust region constraint prevents destructive updates. PPO also learns effectively but shows more variability than TRPO, demonstrating how its simpler "clipped" objective provides stability without the rigid guarantees of TRPO.

2. Acrobot-v1

In this task, the agent must apply torque to swing a two-linked arm up to a target height. The reward is sparse, making it a difficult exploration problem.

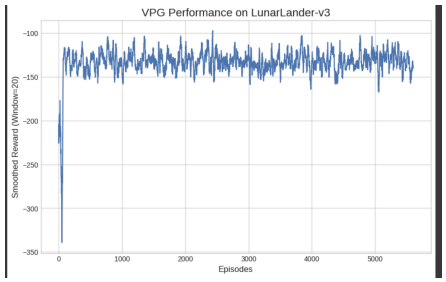
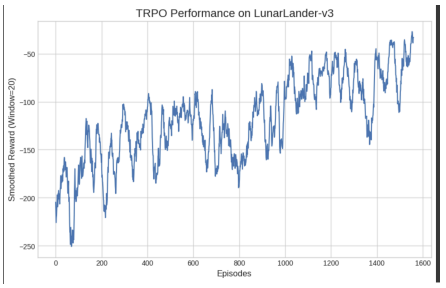
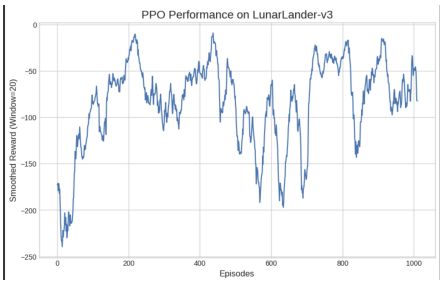
VPG	TRPO	PPO-clip



Analysis: This environment clearly highlights the limitations of VPG. Due to the sparse rewards (a constant -1), VPG completely fails to learn, as its random updates never discover a successful strategy. In contrast, both TRPO and PPO solve the task efficiently. Their policy update mechanisms are stable enough to learn from the noisy reward signal, allowing them to consistently find the optimal policy.

3. LunarLander-v3

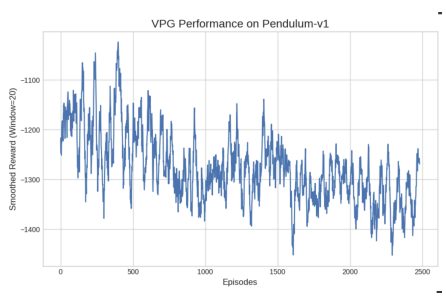
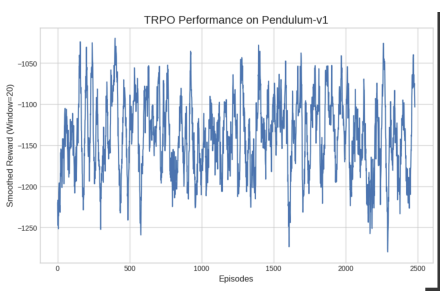
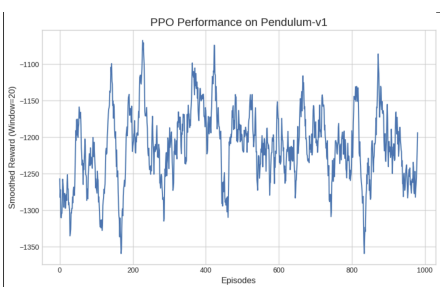
The agent must control a lander to a soft landing on a designated pad, which requires precise control over multiple thrusters.

VPG	TRPO	PPO-clip
		

Analysis: On this more complex task, VPG gets a sudden improvement, which can be by chance and will not be reproducible, after which there was no improvement. TRPO demonstrates slow but steady learning, showing a clear upward trend in rewards, though with significant variance. PPO exhibits the strongest performance, learning much faster than TRPO and achieving a better final reward.

4. Pendulum-v1

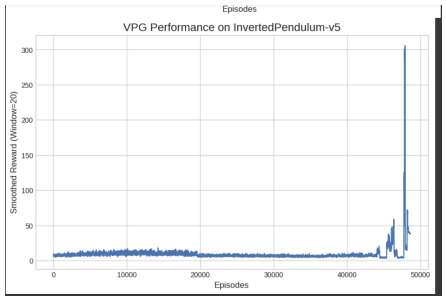
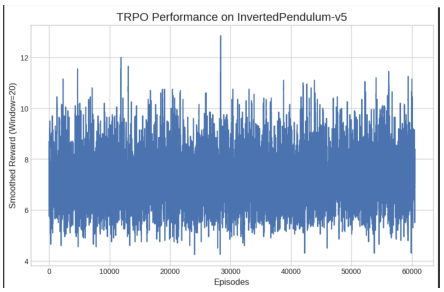
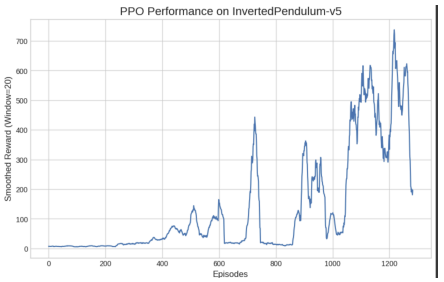
This is a classic continuous control task where the goal is to swing a pendulum up and keep it balanced, minimizing energy usage.

VPG	TRPO	PPO-clip
		

Analysis: VPG's performance is highly erratic and even appears to degrade over time, as its unconstrained updates fail in the continuous action space. Both TRPO and PPO also struggle, showing very noisy and flat learning curves with no clear improvement. This demonstrates that even with the stability improvements of TRPO and PPO, some difficult continuous control tasks require more extensive hyperparameter tuning or more advanced techniques to solve effectively.

5. InvertedPendulum-v5

A continuous control MuJoCo task where the agent must apply force to a cart to keep a pole balanced upright.

VPG	TRPO	PPO-clip
		

Analysis: The results on this MuJoCo task are dramatic. VPG and TRPO both completely fail to learn, with their performance staying flat at a near-zero reward. VPG does hit a lucky spot towards the end, which it did not maintain. They are unable to discover the precise control needed to balance the pole. PPO, however, shows a remarkable ability to solve the task. Its learning curve starts flat but then shows a clear and sustained increase in performance, eventually reaching a high reward.

Conclusion

This comparison shows that while VPG is a foundational algorithm, its performance is limited by the high variance of its policy updates, as discussed in the lecture notes. The trust region constraint in TRPO and the clipped objective in PPO provide crucial stability, allowing them to solve much more complex tasks. Overall, PPO demonstrates the best combination of performance, stability, and implementation simplicity, making it a highly effective and widely used algorithm in modern reinforcement learning.