**REINFORCEMENT LEARNING LAB**

**TOPIC : STABLE BASE IN REINFORCEMENT LEARNING**

* **IMPLEMENTING A CART-POLE GAME**

**SOHAN R. MAYEKAR AARTI SHINDE**

**(2316) (2302)**

**Stable baseline**

**DEFINATON**

A stable baseline typically refers to a point of reference that remains consistent over time, particularly policy-making contexts.

1. **Characteristics**:
   * **Consistency**: The baseline should remain relatively stable over time.
   * **Representativeness**: It should accurately reflect the typical or expected state of the system.
   * **Measurability**: Baselines should be quantifiable and measurable using reliable metrics or indicators.
   * **Validity**: Baselines should be based on sound data and appropriate methodologies.
   * Policy-making: Helps policymakers assess the effectiveness of interventions or regulations.

**Stable baseline uses various reinforcement algorithms**

* **Proximal Policy Optimization (PPO):** PPO is a specific reinforcement learning algorithm used to train agents to learn policies that maximize cumulative rewards in sequential decision-making tasks. PPO is known for its stability and sample efficiency, making it popular for various applications.
* **Deep Q-Networks (DQN):** DQN is another reinforcement learning algorithm that uses deep neural networks to approximate the action-value function. It has been successfully applied to a wide range of tasks, including playing video games and robotic control.

**Importance**:

* **Decision-making**: Baselines provide essential context for making informed decisions and setting realistic goals.
* **Monitoring and Evaluation**: They enable ongoing monitoring of changes and evaluating the success of interventions or policies.

**POLE-CART GAME USING STABLE BASELINE**

In the CartPole game, the stable baseline represents a point where the agent can balance the pole on the cart for a considerable amount of time without it falling. This baseline serves as a reference point for evaluating the agent's performance

* The CartPole problem is a classic reinforcement learning problem where the goal is to balance a pole on a cart.
* The environment is represented by a 2D space where the cart can move horizontally, and the pole is attached to the cart.
* The agent's task is to apply forces to the cart to keep the pole balanced upright for as long as possible.

**Characteristics:**

* Consistency: The agent's behavior should remain relatively stable around the baseline, indicating a consistent level of performance.
* Duration: The agent should be able to maintain the pole's balance for a significant duration, demonstrating robustness and proficiency.

**SETTING UP A CARTPOLE EVNIRONMENT**

Setting up a cartpole environment using the gym library

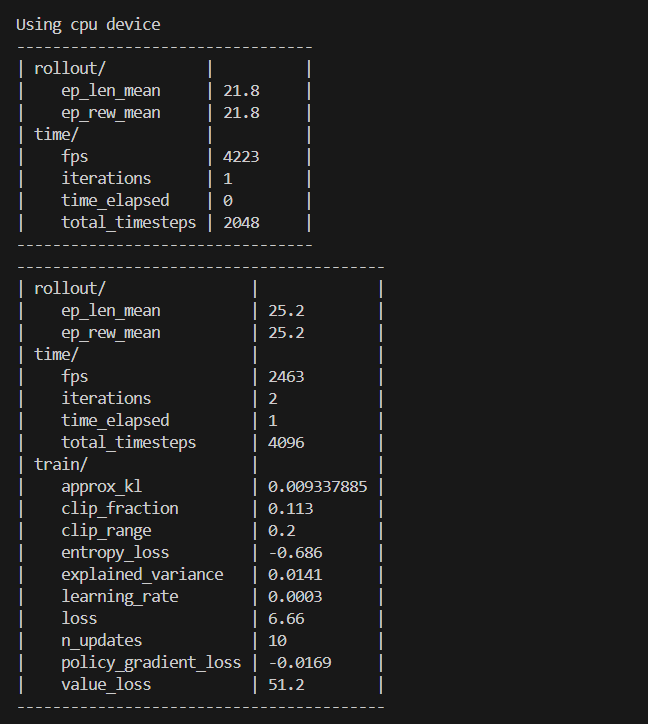
Gym library

Create Environments:

* + Use Gym to create environments representing different tasks or simulations. Environments are defined by Gym's Env class and provide methods for interacting with them, such as reset() to reset the environment to its initial state and step(action) to take an action and observe the next state, reward, and done flag.

**TRAINING**

* **Basic Baseline**: Achieving a stable baseline might involve training the agent using a simple policy such as random actions or basic heuristic rules.
* Here , Proximal Policy Optimization (PPO), is used to achieve higher stability and performance.
* **"MlpPolicy"** specifies that you're using a multi-layer perceptron (MLP) policy network.
* **env** is the environment you created earlier.
* **verbose=1** indicates that you want to see training progress information.



**Training Metrics**:

* + **approx\_kl**: The approximate Kullback-Leibler (KL) divergence between the new and old policy distributions. It measures the difference between the updated policy and the previous policy.
  + **clip\_fraction**: The fraction of policy gradient updates that were clipped. Clipping helps to stabilize training by limiting the magnitude of policy updates.
  + **clip\_range**: The threshold used for clipping policy gradients.
  + **entropy\_loss**: The entropy loss of the policy. Higher entropy encourages exploration by keeping the policy distribution more uniform.
  + **explained\_variance**: The proportion of variance in the returns explained by the value function. A value close to 1 indicates that the value function accurately predicts returns.
  + **learning\_rate**: The learning rate used for updating the policy parameters.
  + **loss**: The total loss incurred during training.
  + **n\_updates**: The number of policy updates performed.
  + **policy\_gradient\_loss**: The loss associated with the policy gradient updates.
  + **value\_loss**: The loss associated with the value function updates.

**TESTING**

Load the pre-trained PPO model for the CartPole problem and then evaluating its performance using the **evaluate\_policy** function.

Evaluate\_policy

evaluating a policy refers to assessing the performance of an agent's learned policy in a given environment.

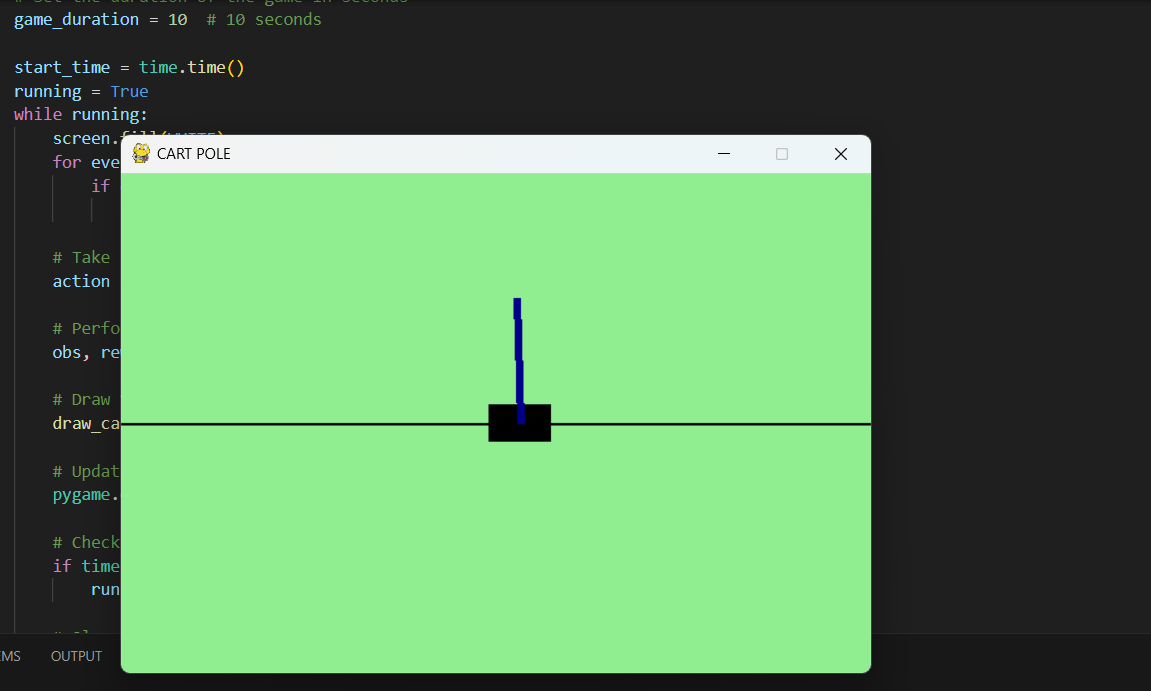
After evaluating the policy the tested agent gives the mean cumulative reward.

****

**PLAY THE CART POLE**

The trained agents now balances the pole without any human or any other interaction

It performs based on how it was trained

****

**REFERENCES**

* <https://github.com/SwamiKannan/CartPole-using-Stable-Baselines/blob/main/CartPole_multi.ipynb>
* <https://stable-baselines3.readthedocs.io/en/master/>
* <https://youtu.be/qMGK5l2vf9w?si=MLLlzgMK04Ha0wJe>