Week 6 Assignment: PPO Experimentation

Candidate: Sneha Santha Prabakar

Coding platform: Google Collab

Please Note:

The python notebook included in the submission contains all the charts and steps mentioned in the grading criteria. This PDF document only provides a summary for each section executed in the python notebook. Hence, please refer to the python notebook for the execution steps.

Part 1: Overview

In order to run each experiment sequentially, and display and compare their outputs in the ipynb notebook, we will make the following modifications regarding the structure of the code from the original Keras version:

- Structure of the original code in the keras website:
  - Import libraries
  - Functions and class
  - Hyperparameters
  - Initializations
  - Train
- Modified structure for the purpose of this assignment:
  - Import libraries
  - Class and only 1 function - discounted_cumulative_sums(). The remaining functions are encompassed in the run_model() function explained below.
  - Hyperparameters (other than *epochs*, *hidden_sizes* and *clip_ratio*)
    - All hyperparameters, except epochs, hidden_size and clip_ratio (that are updated in the experiments), are set as environment variables here - accessible to all functions and operations.
  - Plot function
  - run_model() function that contains:
    - all functions (including the tf functions)
    - initializations
    - train model
    - display output and plot the results

The reason we have re-structured the code to put everything into a single function run_model() is to allow us to efficiently pass parameter values to the model and run different experiments sequentially. We can then view all of their outputs in the same ipynb notebook and easily compare them.

All the code is exactly the same as what was present in the Keras website given in the assignment instructions. The only new code added is for the plot_experiment() function to plot the results of each experiment.

Part 2: Run_model() function

We define a new function called run_model() which basically:

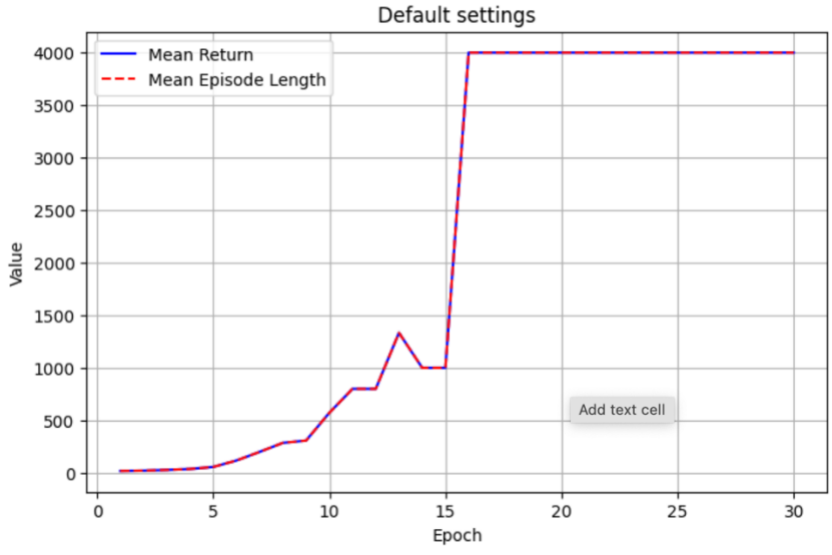
- Retains all the functions and operations already defined in the Keras code
- Allows us to pass updated values of the parameters (epochs, clip_ratio, hidden_sizes) to these functions - as we run each experiment
- Collects the output values (mean return and mean length) and displays it in a plot, for each experiment - allowing us to visualize the trend across the epochs

The function takes 4 inputs (that are updated in each experiment):

1. Epochs: Default value = 30, unless specified in the experiment
2. Clip ratio: Default value = 0.2, unless specified in the experiment
3. Hidden sizes: Default value = (64, 64), unless specified in the experiment
4. Experiment name: We need this for the title of the mean return/mean length vs epochs line plot, for each individual experiment

Part 3: Experiment 0 – Default settings

```
Epoch: 1. Mean Return: 18.433179723502302. Mean Length: 18.433179723502302
Epoch: 2. Mean Return: 21.978021978021978. Mean Length: 21.978021978021978
Epoch: 3. Mean Return: 27.972027972027973. Mean Length: 27.972027972027973
Epoch: 4. Mean Return: 37.38317757009346. Mean Length: 37.38317757009346
Epoch: 5. Mean Return: 56.33802816901409. Mean Length: 56.33802816901409
Epoch: 6. Mean Return: 117.6470588235294. Mean Length: 117.6470588235294
Epoch: 7. Mean Return: 200.0. Mean Length: 200.0
Epoch: 8. Mean Return: 285.7142857142857. Mean Length: 285.7142857142857
Epoch: 9. Mean Return: 307.6923076923077. Mean Length: 307.6923076923077
Epoch: 10. Mean Return: 571.4285714285714. Mean Length: 571.4285714285714
Epoch: 11. Mean Return: 800.0. Mean Length: 800.0
Epoch: 12. Mean Return: 800.0. Mean Length: 800.0
Epoch: 13. Mean Return: 1333.3333333333333. Mean Length: 1333.3333333333333
Epoch: 14. Mean Return: 1000.0. Mean Length: 1000.0
Epoch: 15. Mean Return: 1000.0. Mean Length: 1000.0
Epoch: 16. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 17. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 18. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 19. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 20. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 21. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 22. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 23. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 24. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 25. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 26. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 27. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 28. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 29. Mean Return: 4000.0. Mean Length: 4000.0
Epoch: 30. Mean Return: 4000.0. Mean Length: 4000.0
```



In the CartPole model,

- In each epoch, the agent collects 4,000 time steps worth of experience.
- Reward = +1 for each step survived (where it is able to balance the pole)
- An episode:
    - Begins when the agent starts from the beginning of the environment
    - Ends when:
        - The poll falls - indicated if the pole's angle exceeds a certain threshold; e.g. +- 12 degrees from vertical)
        - The cart moves too far from the center
        - It reaches the max steps (4,000) - Then it did achieved the goal!

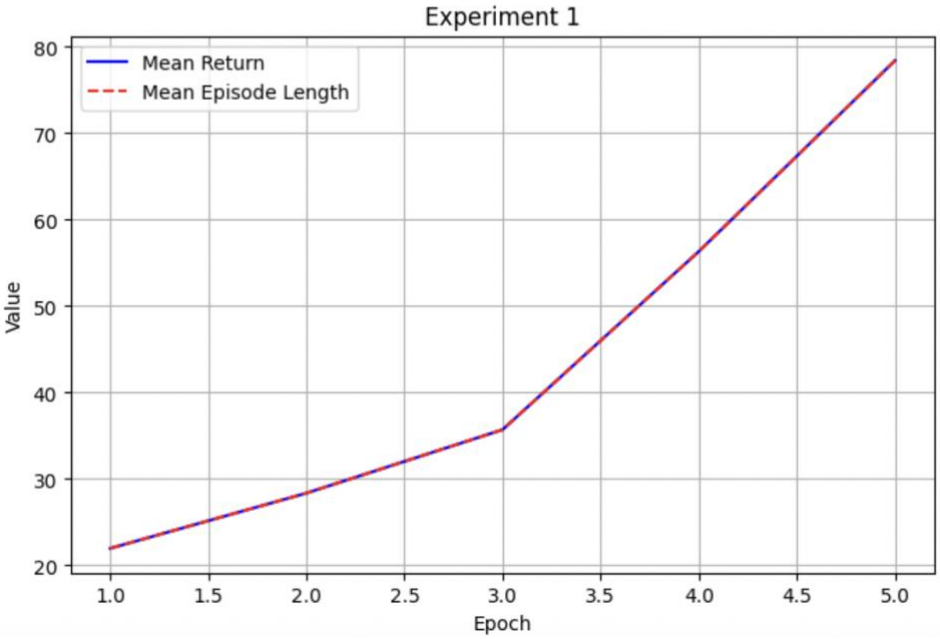
Mean return = Average reward per episode (higher=better)
Mean length = Average number of steps survived per episode (higher=better)

From the results above, we can observe that:

- The mean returns is approx equal to the mean length. This is because in CartPole, each time step = +1 reward (as there are no other reward metrics, like bonuses or penalties).
    - So, the total reward over the episode equals the number of timesteps the pole stayed balanced.
    - Which means, mean return per episode = mean episode length
- Epochs 1–5: Initial Exploration & Random Policy Behavior
    - The policy is nearly random, leading to low mean return (approx 18–56).
    - The agent receives minimal reward since the pole falls early.
    - PPO samples trajectories and collects states, actions, and rewards to estimate advantages.
    - Early updates are small due to conservative clipping (clip ratio = 0.2), preventing large shifts in policy behavior.
    - This phase reflects the agent learning very basic control to increase episode duration.
- Epochs 6–13: Acceleration Phase – Advantage Estimation and Policy Trust Building
    - Mean return increases from approx 117 to over 1000.
    - The policy begins to identify causal relationships between actions and long-term survival (e.g., centering the cart, resisting pole deviation).
    - Generalized Advantage Estimation (GAE) becomes more meaningful due to longer trajectories.
    - With more informative feedback, policy gradients become more accurate, leading to improved updates within the trust region.
- Epochs 14–16: Breakthrough and Sharp Policy Improvement
    - A dramatic jump occurs at Epoch 16: return leaps from 1000 → 4000.
    - Likely due to the agent discovering a near-optimal strategy through a high-advantage trajectory that reinforces survival behavior.
    - PPO's clipped objective avoids overshooting while still updating the policy enough to generalize well.
- Epochs 17–30: Convergence and Plateau
    - Return and episode length remain constant at the maximum value (4000) - which is also the maximum reward.
    - Agent has converged to an optimal policy, balancing the pole indefinitely (until the max episode limit).
    - At this stage, policy entropy is low - the agent is highly deterministic and exploits learned behavior with minimal exploration.
    - Policy updates have little effect due to lack of variance in returns - training becomes a formality.

Part 4: Experiment 1

```
Epoch: 1. Mean Return: 21.978021978021978. Mean Length: 21.978021978021978
Epoch: 2. Mean Return: 28.368794326241133. Mean Length: 28.368794326241133
Epoch: 3. Mean Return: 35.714285714285715. Mean Length: 35.714285714285715
Epoch: 4. Mean Return: 56.33802816901409. Mean Length: 56.33802816901409
Epoch: 5. Mean Return: 78.43137254901961. Mean Length: 78.43137254901961
```



Epochs: 5

Clip Ratio: 0.2

Hidden Size: (64, 64)

Return at Epoch 5: 78.43

- Epochs 1–2: Initial Exploration
  - Mean return increases from approx 22 to approx 28.
  - The agent is beginning to form basic understanding of which actions prevent failure.
  - PPO collects short, mostly unsuccessful episodes, so advantage estimates are noisy and unstable.
- Epochs 3–5: Policy Stabilization
  - Gradual increase to approx 78 return by Epoch 5.
  - Agent now generates longer episodes, enabling better credit assignment through bootstrapped returns.
  - PPO's clip ratio (0.2) ensures that updates are stable but small - prioritizing reliability over speed.
  - Still far from convergence; no clear generalization yet - the agent hasn't optimized yet.

Report question: Is the agent improving meaningfully by epoch 5?

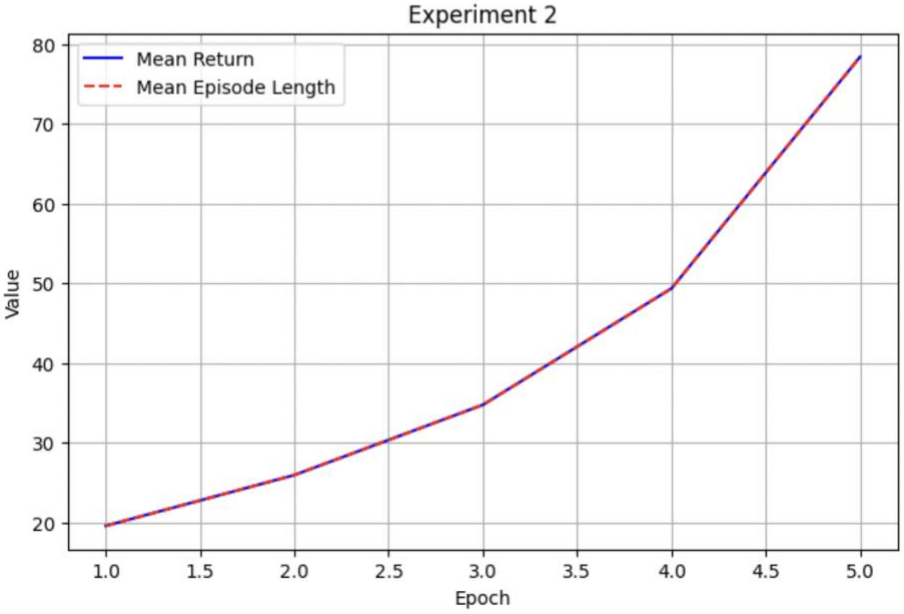
Yes, meaningful improvement is happening, but the policy is still in its early learning phase.

The mean return increases steadily from approx 22 to approx 78, indicating that the policy is learning to survive longer in the CartPole environment. This is a clear upward trajectory, showing that PPO is successfully reinforcing positive behaviors. However, it is still early in training - the agent hasn't discovered optimal control yet, and return is far below the environment's maximum.

The improvement is meaningful for an initial 5-epoch window, but more epochs are needed for convergence.

Part 5: Experiment 2

```
Epoch: 1. Mean Return: 19.607843137254903. Mean Length: 19.607843137254903
Epoch: 2. Mean Return: 25.974025974025974. Mean Length: 25.974025974025974
Epoch: 3. Mean Return: 34.78260869565217. Mean Length: 34.78260869565217
Epoch: 4. Mean Return: 49.382716049382715. Mean Length: 49.382716049382715
Epoch: 5. Mean Return: 78.43137254901961. Mean Length: 78.43137254901961
```



Epochs: 5

Clip Ratio: 0.2

Hidden Size: (128, 128)

Return at Epoch 5: 78.43

- Epochs 1–3: Same Pattern as Experiment 1
    - Despite the larger neural network, return growth mirrors Experiment 1.
    - A wider model provides higher capacity for function approximation, but in early training, this isn't fully utilized because:
        - PPO is still collecting short episodes.
        - Few high-quality examples for deep networks to extract patterns from.

- Epochs 4–5: Plateau Similar to Experiment 1
    - No substantial gain over smaller network.
    - Possibly suffering from under-utilization of model capacity or insufficient updates to leverage deep architecture.
    - Larger models may help later in training, when finer representation is required, but don't necessarily help early unless paired with deeper optimization.

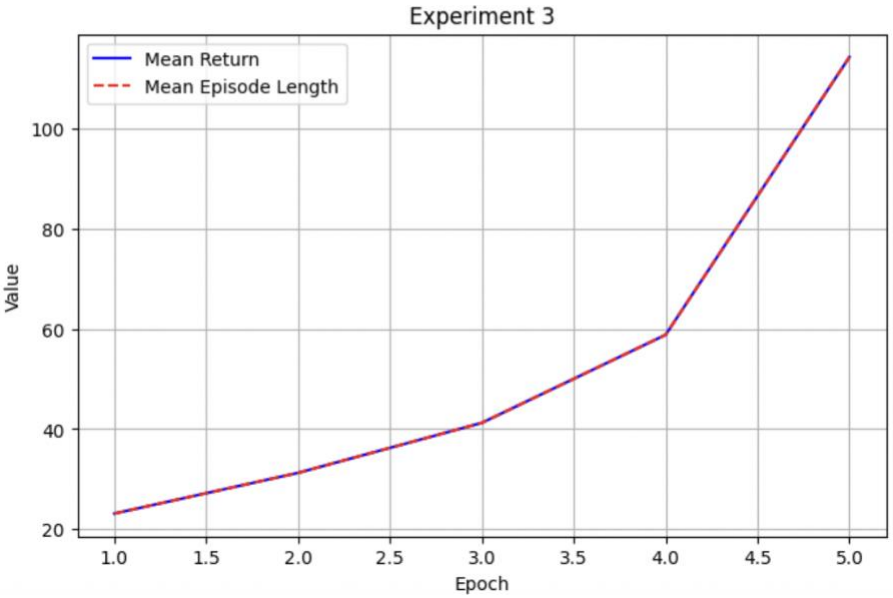
Report question: Any noticeable improvement in convergence speed or return?

No, the larger network does not show any early advantage in convergence speed or return.

- The return trajectory (from approx 20 to approx 78) is almost identical to Experiment 1 (which used (64,64)).
- In early PPO training, the advantage of higher model capacity may be underutilized due to short episodes and sparse reward signals.
- Larger networks might offer benefits later in training (e.g., better generalization, policy expressiveness), but within 5 epochs, there's no gain in learning speed or performance.

Part 6: Experiment 3

```
Epoch: 1. Mean Return: 23.121387283236995. Mean Length: 23.121387283236995
Epoch: 2. Mean Return: 31.25. Mean Length: 31.25
Epoch: 3. Mean Return: 41.23711340206186. Mean Length: 41.23711340206186
Epoch: 4. Mean Return: 58.8235294117647. Mean Length: 58.8235294117647
Epoch: 5. Mean Return: 114.28571428571429. Mean Length: 114.28571428571429
```



Epochs: 5

Clip Ratio: 0.4

Hidden Size: (64, 64)

Return at Epoch 5: 114.28

- Epochs 1–2: Initial Behavior
  - Starting return slightly higher than others (approx 23).
  - Policy updates are more aggressive because clip ratio is 0.4 - allowing larger trust region violations.
  - Larger updates can accelerate learning, but may increase risk of instability.
- Epochs 3–5: Faster Return Growth
  - By Epoch 5, return hits approx 114, approx 45% higher than Experiments 1 & 2.
  - Faster learning occurs because the policy can shift more drastically in response to positive feedback (higher advantage trajectories).
  - However, with aggressive updates, variance in returns may grow in longer runs.

Report question: Is learning faster or more unstable? Any signs of early stopping?

Yes, learning is faster; no instability or early stopping is seen so far.

- Return increases from 23 to 114, which is significantly higher than Experiments 1 & 2 (78 at epoch 5).
- This faster learning is due to the higher clip ratio (0.4), which allows larger policy updates, leading to quicker adaptation.
- The learning curve is smooth and monotonically increasing - no dips or erratic fluctuations are visible yet.
- While high clip ratios can become unstable over long training runs, no instability is observed within these 5 epochs.
- Also, there is no evidence of early stopping - the return is still far from the environment's max, and the upward trend suggests the agent is still improving.

Part 7: Overall Conclusion

- PPO's stability comes from clipping, but this can slow learning in early epochs - Experiment 3 shows a more flexible clip ratio (0.4) allows faster adaptation.
- Larger models (Experiment 2) don't yield better results early unless training is extended or task complexity increases.
- Experiment 0 (default settings) exemplifies full convergence behavior: slow start, rapid acceleration, then plateau - a healthy PPO learning trajectory.
- Early returns are indicative of initial exploration, not policy convergence - 5 epochs is too short to evaluate policy quality conclusively.