University of Oslo
Department of Mathematics

Fall, 2024
Mandatory Assignment 2

Author: Jonas Semprini Næss

TEK5040 - Deep Learning for Autonomous Systems

Answers and Analysis

Actor & Critic

This task implements an actor-critic architecture. Identify the actor and critic in this system. Name an advantage of having a critic.

Answer:

The actor in reinforcement learning relates to the policy-based methods that the given model is equipped with to instruct the agent on which choices to take. In our case this describes the policy network. Furthermore, should the agent be judged on its actions depending on how rewarding the action seems to be, relative to the main objective. This is usually done through value-based methods, whom correspond to the value network.

During training it is essential to get a broad and varied interpretation of the search landscape, especially when there can be optimal solutions that are nuanced (small tweaks in the proposed action). Through having an actor one can thus increase the number of proposed solutions given the magnitude of possible actions. However, when you increase the number of possible solutions and actions the corresponding number of estimated gradients increase drastically, resulting in high dispersion or variance. This in turn can make training less efficient, while also proving to limit optimisation due to cases of local minima in highly variative landscapes. The critic helps mitigate this by evaluating the action and yielding feedback based off of the expected cumulative reward in a state s.

Observation and Action Sampling

Write down the number of the line (in ppo.py) where

- *The action probabilities are calculated*
- *An action is sampled based on the probabilities above*
- *New state (same as an observation in this task) is generated based on the action*

Answer:

- **Line 37**: The function `policy_network.policy(observation)` converts observations into logits, representing raw probability scores.
- **Line 39**: An action is selected by sampling from a categorical distribution based on these logits using `tf.random.categorical`.
- **Line 47**: The environment processes the action via `env.step(action)`, producing the next observation, reward, and additional state information.

Linear vs non-linear policy

Assume that the state of the policy is just the last observation o_t (as has been the case for this exercise). Can you give an example state with which a linear policy can work satisfactorily? Do you expect the optimal policy to be linear?

Answer:

A linear policy in our given environment could work in simple scenarios where there is a direct relationship between pixel data and actions. For example, if the top of the screen is green (grass) and the left side is gray (road), turning left might be a reasonable action. Likewise, if all the surrounding pixels are labelled as road then the clear directional cue is too probably increase velocity in which a linear policy would manage well.
However, the optimal policy likely is not linear. More complex situations such as needing to brake based on both speed and the proximity of turns, require non-linear interactions. Therefore, while a linear policy might handle basic tasks, it would struggle with more intricate dynamics like managing the velocity and multi-variable decision-making like curvature, distance boundaries and sharpness of turns.

Value function example

Give an example of a state, i.e. (o_t, time_remaining) which has a high value function (high average return). Can you think of a state having a low value function.

Answer:

A high-value state, (o_t, `time_remaining`), would involve the car moving efficiently along the **center** of the track with sufficient time left, and moderate to high speed, provided that the track also extends straight ahead. Conversely, a low-value state could involve the car skidding off-track or being stuck in an area/region outside the track with minimal remaining time.

Policy and value network architectures

Based on the given implementation, describe the architectures of the policy network and the value network (i.e. state what type of layers they have including their dimensions. State also the intended dimensions of input and output of these networks.)

Answer:

Policy Network Architecture

- **Input:** Observations from the environment, $(batch\_size, height, width, channels)$:

  Input Dimension for Policy Network: $(1, 96, 96, 3)$

- **Layers:**
  1. **FeatureExtractor:**
     - **Convolutional Layers:**
       * Conv2D (16 filters, kernel size 8, stride 4) $\rightarrow$ Output: $(batch\_size, 24, 24, 16)$
       * Conv2D (32 filters, kernel size 4, stride 2) $\rightarrow$ Output: $(batch\_size, 12, 12, 32)$
     - Flatten Layer $\rightarrow$ Output: $(batch\_size, 12 \times 12 \times 32)$
  2. **Dense Layer:** Outputs logits for each action (dimensions: $(batch\_size, num\_actions)$).

- **Output:** Discrete action indices, shape $(batch\_size)$.

Value Network Architecture

- **Input:** Observations (same as the policy network) plus additional information like `time_left`.

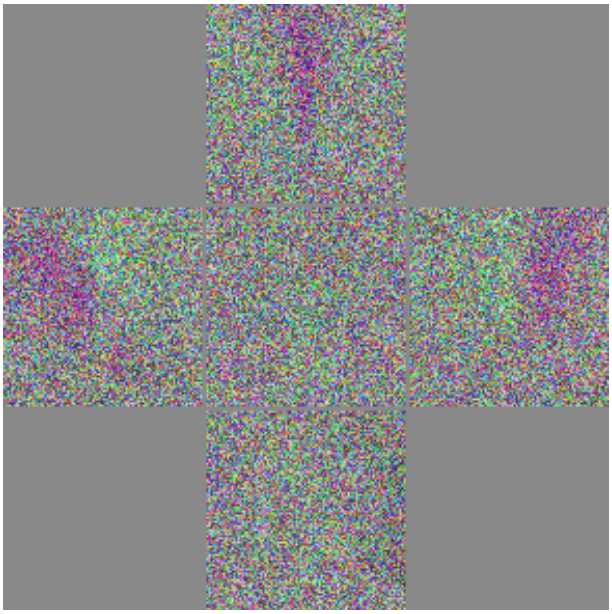
- **Layers:**
  1. **FeatureExtractor** (as in the policy network).
  2. **Concatenation Layer:** Combines feature output and `time_left` (dimension adjusted).
  3. **Optional Dense Layer:** Outputs to a single value.
  4. **Dense Layer:** Outputs value estimation (dimensions: $(batch\_size, 1)$).

- **Output:** Estimated state value, shape $(batch\_size)$.

Visualization of policy network weights

Save this figure and put in the report. Are you able to interpret the weights in any way?

Answer:



Figur 1: Visual of filter weights.
Top: Gas, Right: Right-turn, Left: Left-turn, Bottom: Brake

The weight distribution for the left-turn filter resemble a pattern suited for detecting a left turn, which fits well, and seems reasonable. A similar distribution can be seen for the filters for right-turn and gas. With more training, different optimal policies and hyper-parameter tuning, the visual might become even clearer.

Eval policy

Evaluate your model for the highest scoring iteration (in terms of mean). Report scores for <N> equal to 1, 2, 4 and 8. Report both minimum, median, mean and maximum scores for all cases. How would you judge its performance qualitatively?

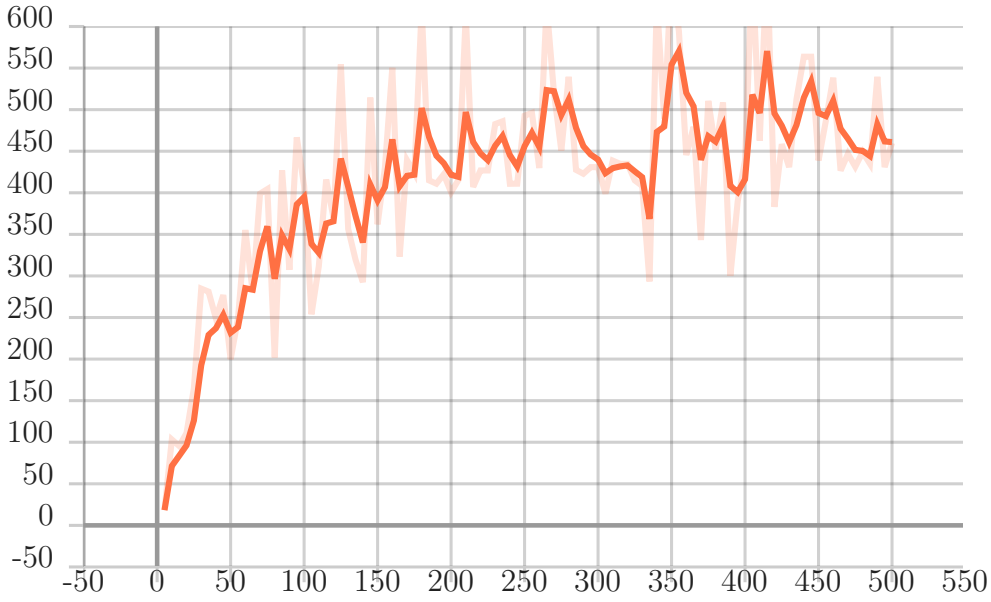
Answer:

N	Min	Max	Median	Mean
1	70.9091	224.755	145.822	148.842
2	92.8527	266.853	161.191	167.016
4	91.1111	254.131	171.428	171.379
8	62.7068	192.121	133.045	133.691

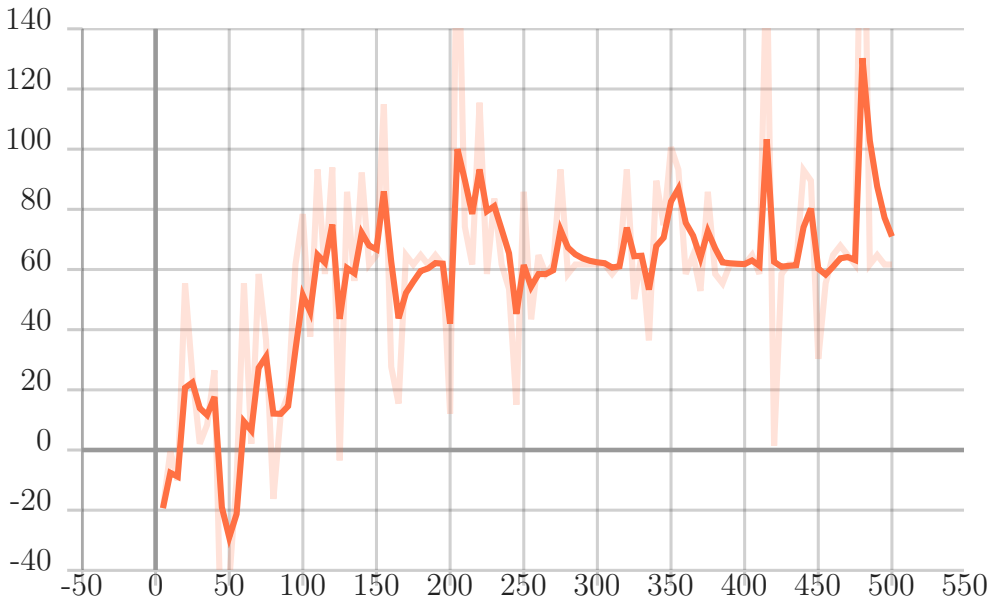
Tabell 1: Summary Statistics for Different action-repeat values

For action-repeat values $N = 1, 2, 4$, and 8, the model peaks in mean performance at $N = 4$ (mean = 171.379) and shows stable medians across all values. Indicating that, less action-repeats can yield better and more optimal paths/trajectories.

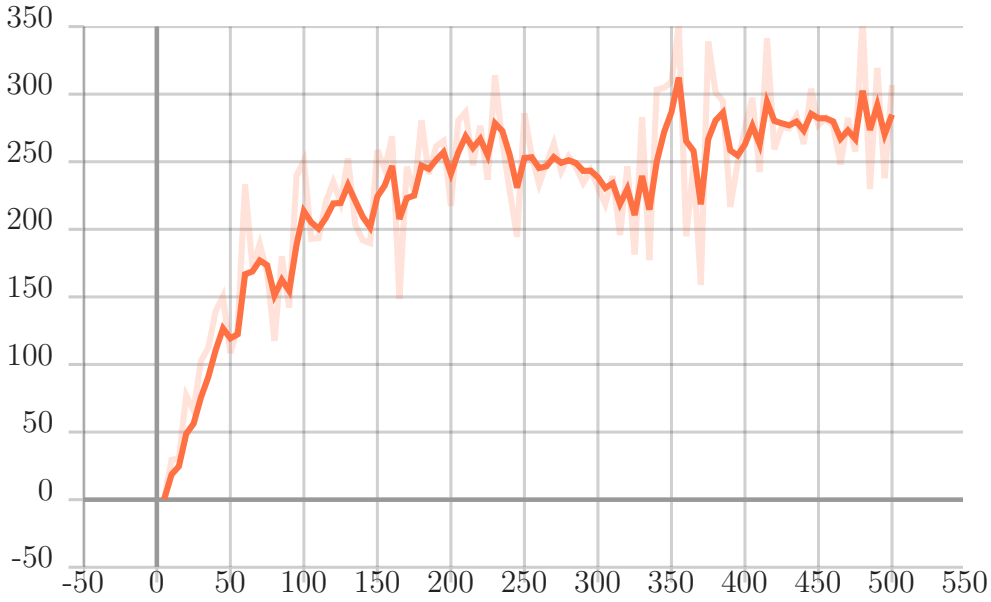
Plots



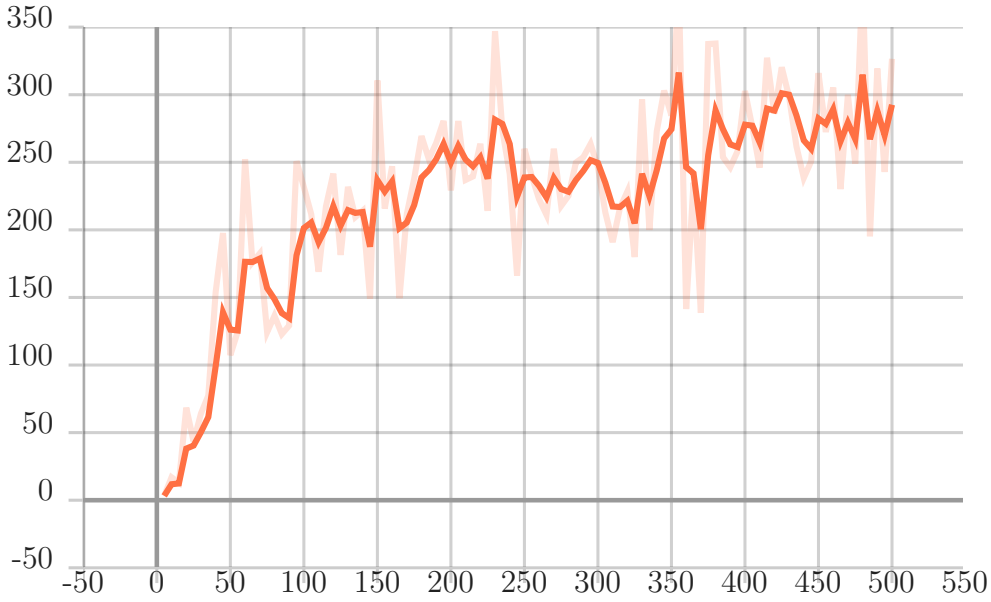
Figur 2: Return Max values



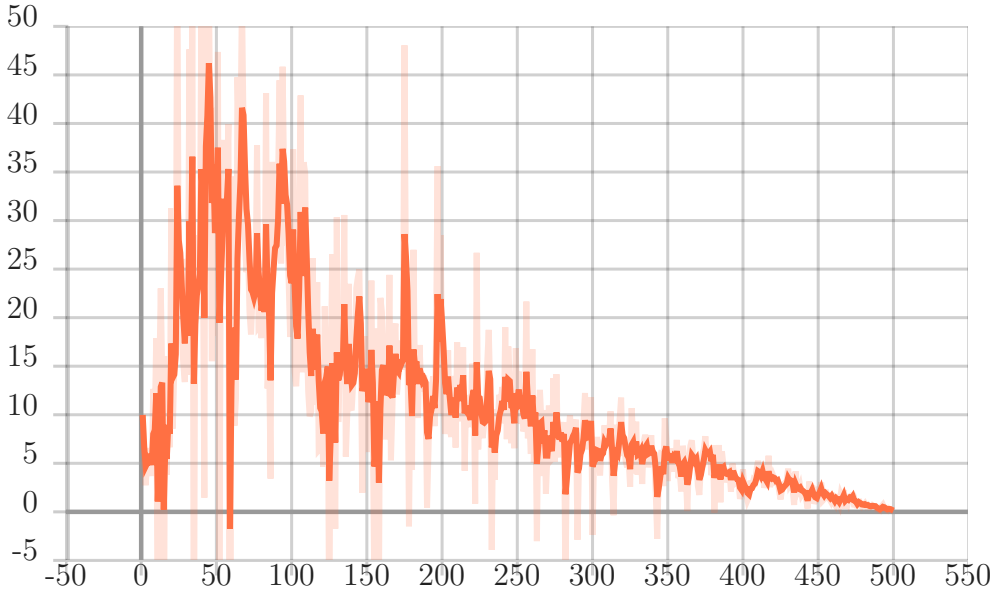
Figur 3: Return Min values



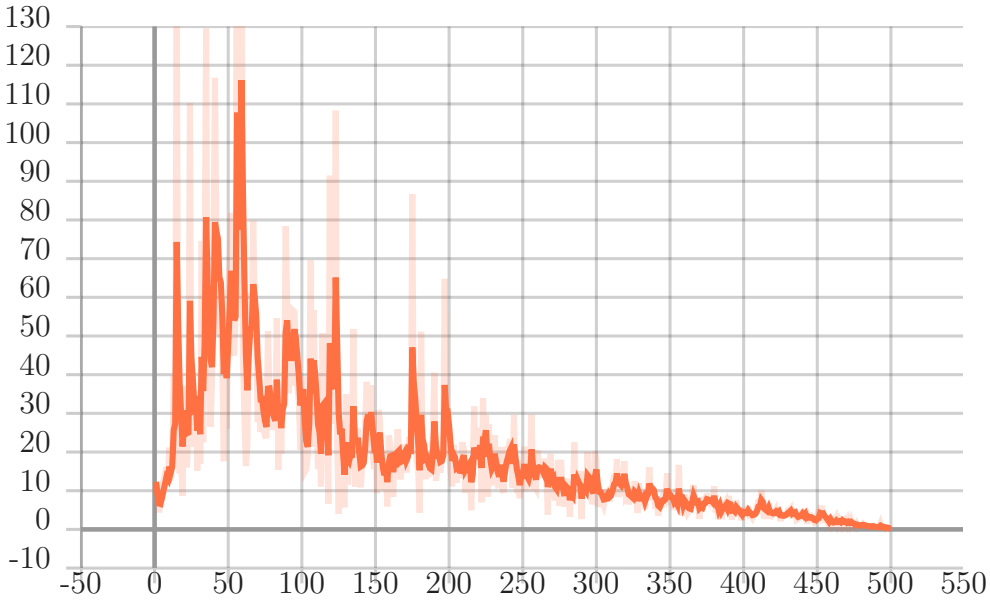
Figur 4: Return Mean values



Figur 5: Return Median values



Figur 6: Estimated Improvement



Figur 7: Estimated Improvement lb