Short summary of the exercise topic

In this first assignment, we have to develop a deep reinforcement learning model that can learn how to perform actions (i.e., driving) in a simulated environment (i.e., street).

The name of the environment is “CarRacing-v2” provided by OpenAI’s Gymnasium and the goal is to make real time decision to drive a car with one of the following actions: gas, brake, steering (left, right), and nothing.

The exercise is to create a neural network that can understand the current state, anticipate future states, and generate a real-time action.

How you solved the problem

For the network architecture, I used:

- Two convolutional layers (the first with 16 filters and the second with 32 filters) to extract the feature, both with kernel 5 and stride 1
- Max Pooling Layer after each convolutional layer to reduce the dimension by half.
- Fully connected layers that output the logits for each possible action
- Forward pass:
    - the forward function modifies the input according to the network design.

For the training environment, I used:

- Training Loop: the training is done with cross-entropy loss function and Adam optimizer. The train is performed by looping over batches (with frames and actions) executing a forward pass, computing the loss, and updating the weights.
- Evaluation (with provided Logger Class): training and evaluation losses, and accuracy are set for the Logger class and therefore are stored.

For the dagger function, I used:

- A beta to select which policy use, which decreases over iterations, gradually shifting the policy away from the expert. This means that the model will initially learn from the expert and then rely on its own policy decision.
- A while loop based on the parameter “done” that comes from the env. It will go out of the loop when I will be turned to True (car crashed)

Performance of your agent in your own evaluation (>725)

The agent performs quite good in a long run, however, small mistakes lead to some off-tracks. To correct those off-tracks the agent steers more, and the amplitude of the oscillations increases until the car is completely off-track. The agent then makes a full-roll, trying to return on track without success.

Maybe a lack of expert cases in this situation is the reason why it gets in trouble with full off-track.