PPO on `highway-v0`: Training an RL Agent

Sarvadnya Nandkumar Purkar
22B4232
Indian Institute of Technology, Bombay

July 16, 2025

Algorithm and Implementation Details

For this project, I chose to work with the PPO (Proximal Policy Optimization) algorithm using the `stable-baselines3` library. PPO felt like a solid choice because it's relatively easy to implement and gives decent results in continuous action environments.

I trained the agent in the `highway-v0` environment from the `highway-env` library. I used the MLP policy and kept most of the default hyperparameters to keep things simple. The environment was configured with 5 vehicles and a short episode duration to speed up training, as running too long on Colab (especially CPU) becomes slow and painful.

A custom callback was added to log episode rewards, and I used Matplotlib to plot the reward curve at the end. The total training was done for 25,000 to 100,000 timesteps, depending on runtime. I also generated a short video using `VecVideoRecorder` to visually verify if the agent had learned anything meaningful.

Results and Metrics

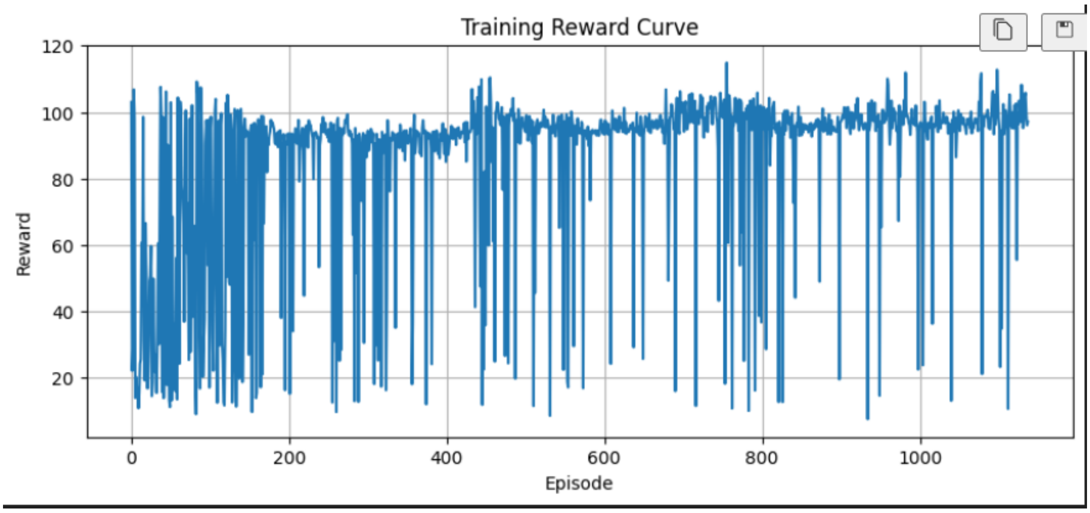

Figure 1: Training reward vs. episodes

As seen in the plot above, the training reward was quite unstable at first, but eventually the agent started getting higher and more consistent rewards. Around the 800–1000 episode mark, the agent was mostly doing fine with occasional drops (probably due to crashes or bad exploration).

The max reward reached was around 100+, which seems to be near the environment's upper limit.

Insights and Challenges Faced

Initially, setting up the environment took some effort. For example, the `configure()` function doesn't work directly on the wrapped env returned by `gym.make()`, so I had to use `highway_env.make()` instead. Also, video rendering using `VecVideoRecorder` was a bit slow, especially on CPU, so I had to reduce the number of frames.

One challenge was debugging reward instability — PPO keeps exploring so the curve never fully flattens. But over time, the average reward was increasing, which was reassuring. Using GPU on Colab helped a lot with reducing training time.

Overall, it was a nice hands-on exercise. I now have a better understanding of how PPO works in practice and how to interact with Gym-like environments for reinforcement learning.