RL Lab 5

Harsh Raj (180010017)
Akhilesh Bharadwaj (180010009)

Note: Code for all questions (including simulations etc) is attached in the submitted zip. Artifacts including the images generated are also attached in the zip. Please make sure to install dependencies mentioned in the requirements.txt before running any submitted code.

```
# Install dependencies
pip3 install -r requirements.txt

# run codes corresponding to each question
python3 runner.py
```

All the charts can be accessed online on this wandb repo.

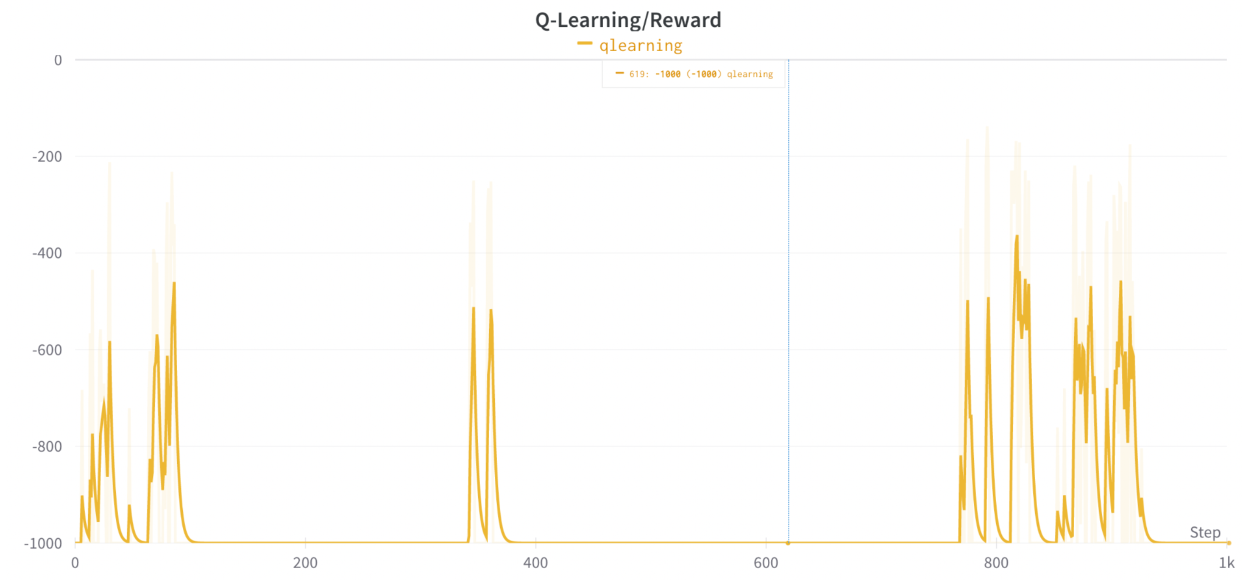
Note that we were not able to complete the following during the given time:

- SARSA - Both
- Q Learning with RBF
- Reinforce - Both
- Also, we could try QLearning with tile coding on only 1 task as most of the time was spent on debugging the Tile coding part.

Q Learning with Tile Coding for Mountain Car

We used 2 tiles to approximate the value functions. The whole state space was divided into a grid of shape (10 x 10). The agent was trained using 0.2 learning rate, and gamma as 0.9
The probability of selecting a random action was started with 0.2 and was linearly reduced to 0.

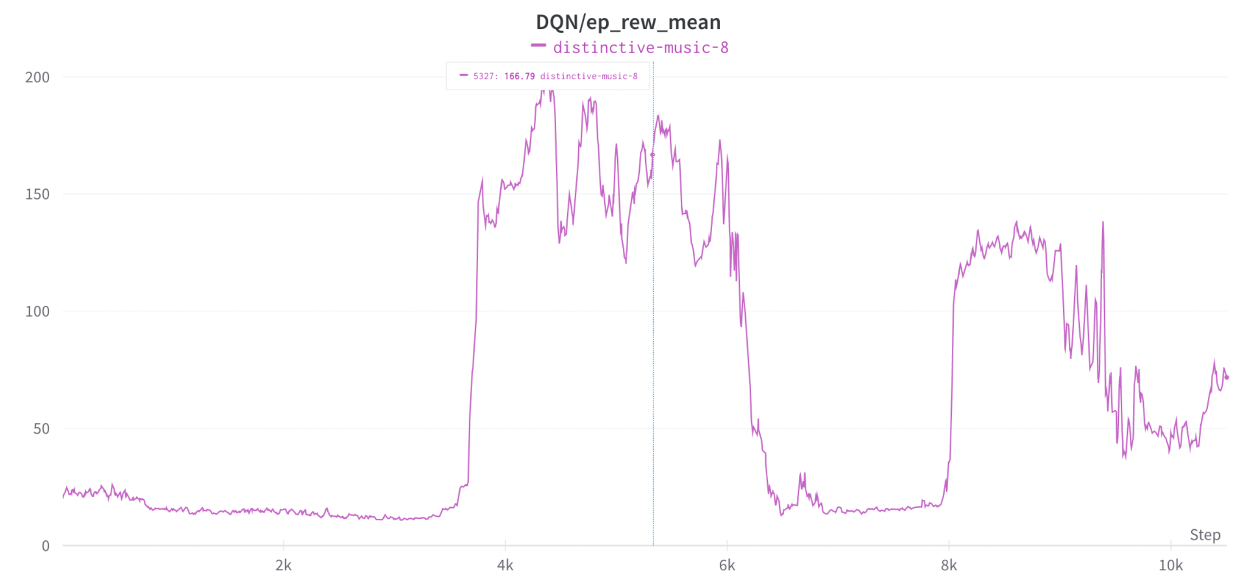
Attached below is the reward graph for the same

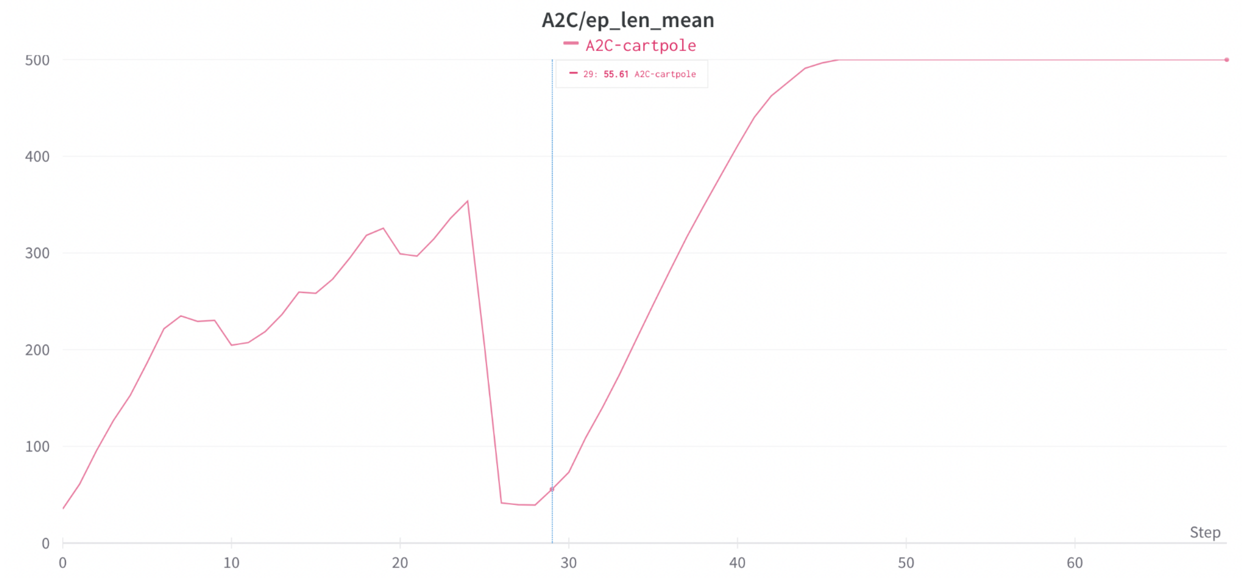


The max timestep of the environment was increased from 200 to 1000.

DQN and A2C on Cartpole

We tried the DQN and A2C algorithm on the standard gym task of Cartpole. We used the stable_baselines3 library with default hyper parameters to train the agents.





We observed that A2C performs well in general compared to DQN.