Hadoukenator

Reinforcement Learning with Proximal Policy Optimization

Mason Stott, Akash Pramod Kumar

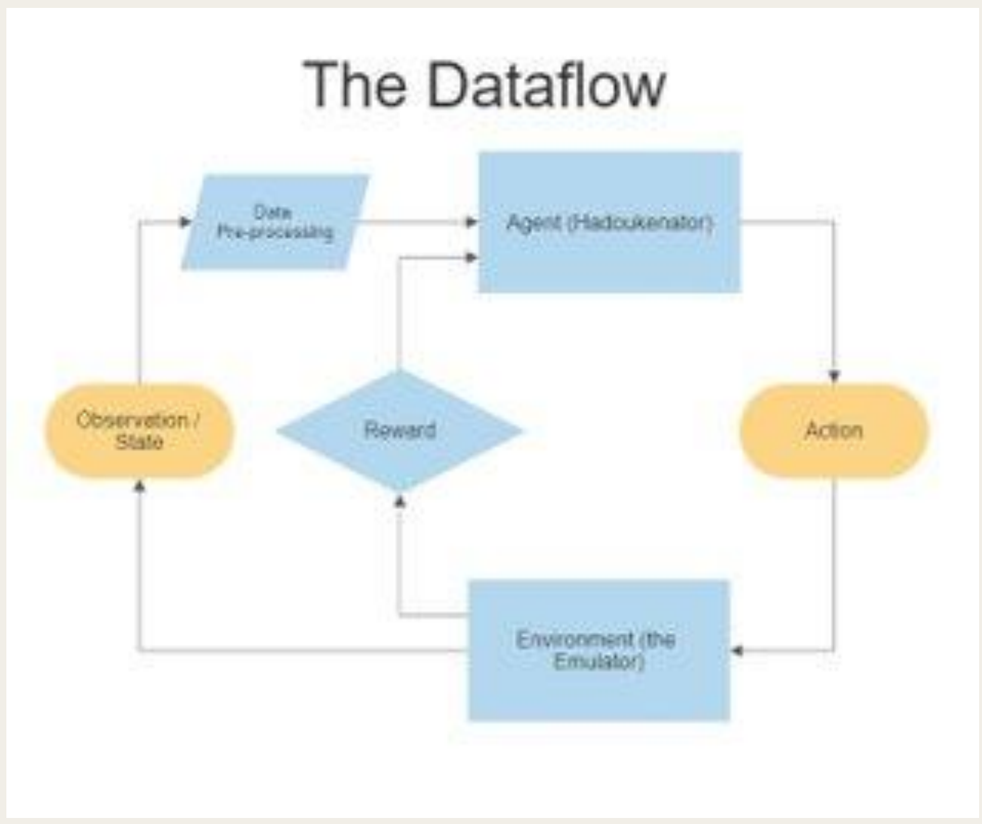
Introduction / Motivation

- Reinforcement learning is well known to be suited for complex tasks in response to comprehensive data, and there are many algorithms available.
- Proximal Policy Optimization (PPO) is a state-of-the-art algorithm known to perform well in continuous control tasks that updates the policy by maximizing an approximation function.
- Video games can be quite complex, and they are fun to watch train, so we decided to evaluate how well the PPO algorithm can perform in Street Fighter 2 with limited training.
- Hyperparameters:
  - Learning rate: controls the step size of the algorithm while searching for the optimal policy.
  - Discount factor: determines the importance of future rewards relative to immediate rewards.
  - Entropy coefficient: controls the degree of exploration by encouraging stochasticity.
  - Clip range: limits the change in policy between updates.

Methods / Approach

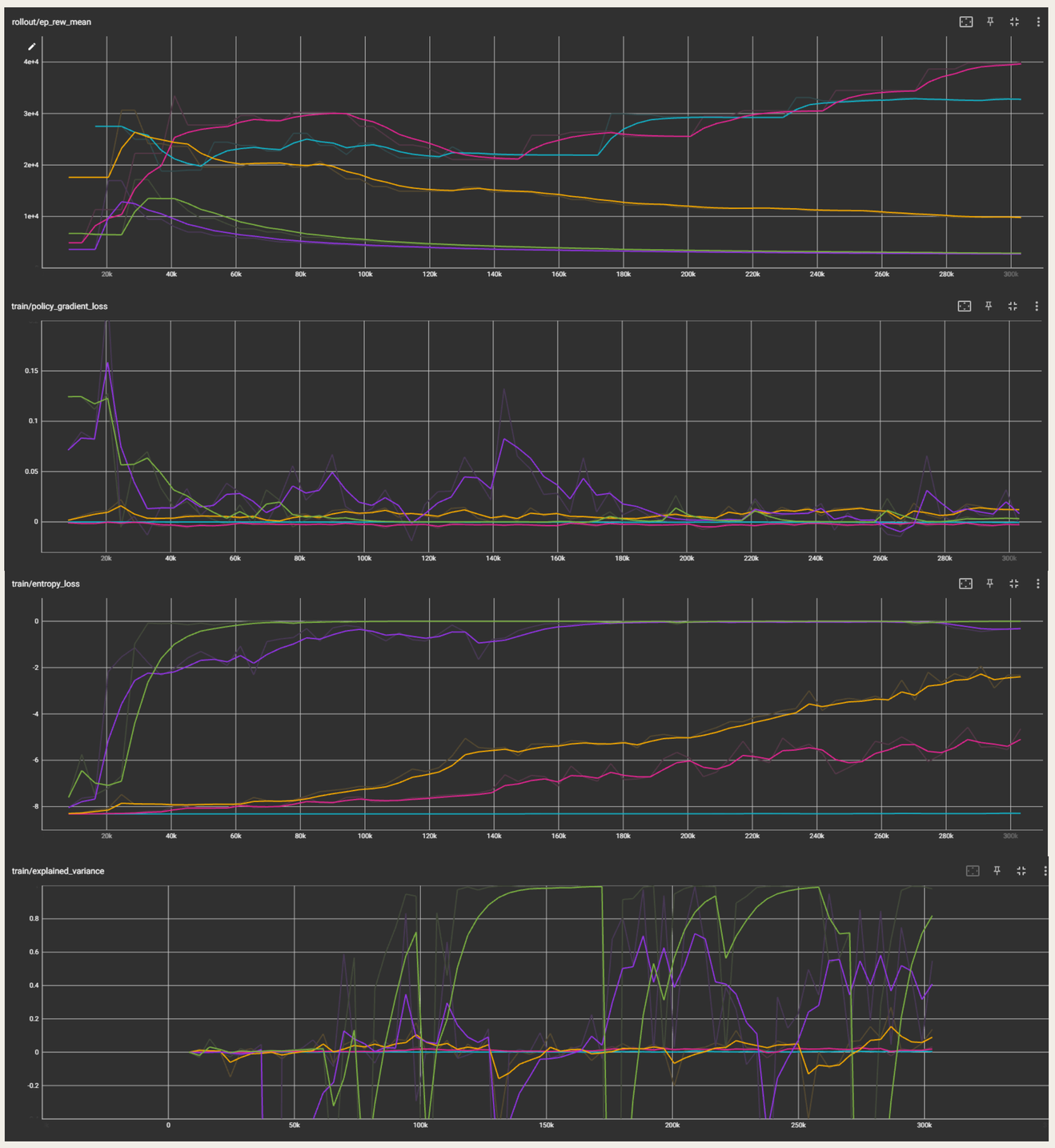
Thankfully, Open AI has many libraries available for our application, such as stablebaselines3 (PPO) and gym-retro (game emulator). These two resources allowed us to proceed smoothly.

- Interface with the game and play as the character Ryu in the story mode.
- Initialize our RL models with varied parameters.
- Preprocess the observations of the model as much as possible to condense training time.
- Fitness function: simply evaluate the final score the agent achieved in the game.



Results

For the scope of our study, the parameter we took the most interest in was that of the learning rate, as it controls how drastically the policy adopts changes in order to “improve.”



Key (Learning Rates):

- **5e-7, 5e-6, 5e-5, 5e-4, 5e-3**

Conclusions

- Determining a suitable learning rate is crucial to preserving performance.
- Without proper hyperparameter tuning, models can quickly deteriorate or “overfit.”
- Condensing the observation space efficiently can see large gains in model size in training time.

Future Work

- Investigate reproducible methods of hyperparameter tuning on a per-application basis.
- Compare PPO to other RL algorithms (namely A2C) on the same applications.
- Compare performance of RL algorithms to a genetic algorithm approach (LEAP).

THE UNIVERSITY OF TENNESSEE KNOXVILLE