Greifer AutoRL

Research Head:

Nik Brown

Email: nikbearbrown@gmail.com

Expected Start Date: 08/07/2020

How to get started:

1. Come to the weekly meetings which will start after September 7, it will be weekly meetings

2. Learn about the basic idea of the project, get involved in the discussion, and get your questions

   answered

3. Start working and get the feedback and constructive criticism to improve it.

Prerequisites:

Experience with **Python**

The "greifer" project build tools that simplify, pipeline and improve the process of reinforcement learning (RL) o Most of the work in RL is preparing the environment, shaping the rewards, and visualizing what an agent is doing while it learns o Complements existing RL libraries. Compliments the course Deep Learning and Reinforcement Learning in Game Engineering - CSYE 7370.

Starting with a book called "The Nuts and Bolts of Deep RL" or something similar.

Projects associated with this research

Griefer Open-Source Project (python library and book)

- Creating environments o Signal selection and engineering
- Shaping and optimally weighting rewards
- Choosing RL algorithms for a given problem
- Initialization
- Establishing baselines
- Early stopping metrics
- Interpreting and visualizing RL policies, agent behavior, environment dynamics, and reward dynamics

Griefer Side Project for GameDevs

- Create a tool the maps RL libraries to Unreal Engine as MLAgents does for Unity. This will be the basis of an Epic MegaGrant https://www.unrealengine.com/en-US/megagrants
- This has the possibility of a paid project if funded. The focus will be writing the Epic MegaGrant. This basically means writing a design document and budget that explains how existing RL libraries can map to Unreal Engine as ML-Agents does for Unity
- The basic need is to create the environment in Unreal and export it so that it can used by TF-Agents, and then running a saved model from TF-Agents so that Unreal can run its action decisions

Basic structure of the Book

Nuts and Bolts of Deep RL (TOC) Series 1

- What is RL?
- Optimal Stopping Problem
- Multi-armed Bandits
- Tabular Solution Methods (Value Based)
- On-policy versus Off-policy
- Policy Gradient Methods
- Environments
- Signal selection and engineering
- Reward shaping and weighting Initialization
- Establishing baselines

Nuts and Bolts of Deep RL (TOC continued) Series 2

- Early stopping metrics
- Signal and Reward
- Normalization
- Choosing DL Architectures
- Choosing RL Algorithms
- TensorFlow Agents Algorithms (DQN, REINFORCE, DDPG, TD3, PPO, SAC) - Value Based versus Policy Based
- RL Visualization
- RL policies
- Agent behavior
- Environment dynamics
- Reward dynamics

Note: Microsoft teams will be the mode of communication for this research. Click here to join the channel.