Draft Reinforcement Learning Short-Course Syllabus

October 18, 2021

Quansight
YOUR DATA EXPERTS

8656 W. Hwy 71, Bld F20,
Austin, Texas

512-354-2975

connect@quansight.com

www.quansight.com

Overview

We propose a short course that could be given to data scientists and other practitioners covering the practical considerations needed to effectively apply reinforcement learning to real-world problems. The course will use a testing environment (Robot Ant) that simplifies some aspects of implementation but addresses real Physics-based learning tasks.

Topics

- Reinforcement Learning concepts and limitations
- Reward function design and shaping
- Hyperparameter tuning
- Application of a trained model

The course would be given in 6, 1.5-2 hour sessions over a number of weeks (probably 3 weeks).

Precourse reading

- openai-spinning up
    - https://spinningup.openai.com/en/latest/index.html
    - Just the background: https://spinningup.openai.com/en/latest/algorithms/td3.html
- Other papers pertinent to TD3 (Read through the TD3 background and find the key papers)
    - https://spinningup.openai.com/en/latest/spinningup/keypapers.html
- https://www.army.mil/article/242079/army_research_leads_to_more_effective_training_model_for_robots

Schedule and Topics

Day 1: Introduction to reinforcement learning, the PyBullet3 Ant RL environment, OpenAi Gym.

- Why these environments are helpful
    - Discuss continuous vs discrete variables
    - Discuss which environments are used to test which kinds of tasks
- Show several easy environment and modeling tasks (e.g., OpenAI Gym)
- Example PPO/SAC algorithm from simple environments, to show training/env render
- Homework: read original TD3 paper (https://arxiv.org/abs/1802.09477)

Day 2: Brief overview of widely used RL algorithms and TD3 specifically; walk through set up.

- Describe algorithms at high level - how to choose each one
- Describe Ant: default goal and reward

- Create conda env
- Start JLab
- Describe imports (why we need them)
- Walk through notebook code and text. Run each cell as the lecture proceeds and discuss what's happening and why it's important
- Homework: clone /Quansight/Practical-RL, set up and run default ant training

Day 3: Extending and experimenting with RL and PyBullet3 to modify the agent.

- Discuss reasons for modifying the agent (reminder: agent is part of the environment)
- Make new goal for Ant: go to coordinate
- Modify the reward to work with new goal
    - Show how we did it and why - let them know our thought process. For homework, they will do the same, but it will be more difficult without us showing them how to do it.
- Homework: Define a new goal for the ant and put it into practice

Day 4: Key concepts of reward shaping and hyperparameter tuning.

- Discuss reward shaping while answering questions about homework. Reinforce their understanding and the importance of a well defined reward.
    - Usually keep reward very general: avoid forcing the agent to behave in a certain way, focus on rewards for goals instead
    - Example: don't reward the agent for how it walks, reward it for achieving a walking goal, like getting to a coordinate.
- Walk through each hyperparameter and what it can potentially do to improve a model
    - evaluation frequency
    - exploration noise
    - batch size
    - discount
    - tau
    - policy noise
    - noise clip
    - policy frequency
- Model evaluation
    - Common techniques and when to apply them
- Homework: Improve the ant goal to increase generalization

Day 5: Testing the trained model in new scenarios different from training.

- Remember that learning/training is the hard part, but ultimately, we need to use the model
- Evaluate the model on several tasks within the expected range of situations, and maybe outside of that range too (walk to coordinate on flat ground vs climb to top of mountain)
- Homework: Prepare for show and tell

Day 6: Show and Tell - everyone shows what their ants can do and we discuss.

- We see each person's model as they describe what they made, why they made it, how they made it, and what it can do.