ECE 590D-001, Reinforcement Learning at Scale

Jay Hineman, Ph.D.

Geometric Data Analytics

2019

Description

This course consist of three parts. The first part will focus on machine learning at scale using modern tools such as Docker, GitLab with CI/CD, cloud computing, and Kubernetes. The second part will focus on reinforcement learning (RL) for single- and multi- agent environments and include topics such as Q-learning, policy gradients, and their deep learning extensions. The third part will combine the first two topics and focus on scaling DeepRL methods to attack large problems such as the Atari-57 benchmark and the StarCraft Multi-Agent Challenge.

Details

- The Syllabus.
- Other resources:
  - Lecture notes
  - Bibliography (including books, articles, lecture notes from other course, projects)
- How I'm approaching this course as an instructor:
  - Collect and distill resources that allow students to understand recent advances in RL. To this end I will make connections with applications.
  - Develop experiments that illustrate the execution of recent advances in RL.
  - I am interested in the technical details (mathematician), but I think intuition about the field as a whole is more valuable.
  - I think problems and their solution is the best way to develop theory.

- How I'd like you to approach this course as a student:
  - Be curious—there will ample additional reading beyond what I can cover in-class.
  - Try things—learning by doing is critical and there many exciting places to apply RL.
  - Realize that the programming and implementation parts are likely as valuable (if not more) than the theory. It's essentially like steady hands in the lab.
- Caveat Lector
  - I try to clarify which statements are my opinions and which I can support by data or proof. *I will not always succeed at this.*
  - These notes are essentially a draft and will like contain typos, errors, and other forms of mis-information. I'd rather they not, so please send corrections and comments to me via email.
  - I will be version controlling the notes via git+github.

Some very brief history (more in later lectures)

- Reinforcement learning is other than you might think; it starts with Bellman in the 1950s (if not Von Neumann). Here the approach was using dynamic programming (DP) to solve exactly (by specifying a policy or value function). DP does not scale well.
- *Reinforcement learning, Approximate dynamic programming, and Neuro-dynamic programming* are all essentially interchangeable and try to solve the same problem, but by approximate policy or value functions.
- A number of advances were made in the 1980s and 1990s, but were limited by computational power. See also the Wikipedia article on *AI Winters*.

Some very brief history (more in later lectures)

- One way to approximate a policy or value function is using an *deep neural network*—this is part of the recent surge of research activity in RL.
- RL will often contain a generative component that can be sampled in a parallel or distributed sense. The price, availability, and user tools for distributed computing have also driven the RL surge.
- As a result of using new approximation methods (or practicality of such methods), new RL updates are currently active research topic.

Where to start ...

- There are some very nice books (see the Syllabus).
- There are also piles of journal articles (bibliography forthcoming).
- **There is also digging into code!**

Spinningup

- Introductory guide to (deep) reinforcement learning written by Joshua Achiam (OpenAI Research Scientist).
  - Gets to the good stuff quick and mixes theory and implementation well.
  - Uses standard tools (python, tensorflow, mpi)
  - Open-ended and flexible—we'll mix it together with other resources.

Docker

- De facto standard for containerization
- Starting place for rapidly building distributed architectures with orchestration like Kubernetes.
- Starting place: containerize requirements and code for spinning up (this will be written formally as a homework assignment).