CSCE 790 - Production Machine Learning Systems

Pooyan Jamshidi

Fall 2018

https://pooyanjamshidi.github.io/teaching/
Office Hours: TBD
Office: TBD

E-mail: pjamshid@cse.sc.edu
Class Hours: TR 2:50pm-4:05pm
Class Room: SWGN 2A24

Course Description

When we talk about Machine Learning (ML), we typically refer to a technique or an algorithm that give the computer systems the ability to learn and to reason with data. However, there is a lot more to ML than just implementing an algorithm or a technique. In this course, we will learn the fundamental differences between ML as a technique versus ML as a system in production. A production-ready ML system involves a significant number of components and it is important that they remain responsive in the face of failure and changes in load. This course covers several strategies to keep ML systems responsive, resilient, and elastic. Machine learning systems are different than other computer systems when it comes to testing, building, deploying, and monitoring. ML systems also have unique challenges when we need to change the architecture or behavior of the system. Therefore, it is essential to learn how to deal with such unique challenges that only may happen when building real-world production-ready ML systems (e.g., performance issues, memory leaking, communication issues, multi-GPU issues, etc). The focus of this course would be primarily on **deep learning systems**, but the principles will remain similar across all ML systems.

Learning Outcomes

1. Explain differences between ML as predictive technique and as a computer system.
2. Describe how a distributed ML system works in production and insight into challenges.
3. Locate technical debt in building ML systems.
4. Employ design strategies and best practices to mitigate technical debt.
5. Incorporate ML-based components into a larger system.

6. State the principles that govern ML systems.
7. Build systems that are more capable, both as software and as predictive systems.
8. Identify systems issues and apply strategies to avoid them in production ML systems.

Course Syllabi

In this course, we will understand central principles of **production machine learning systems**. We will begin by reviewing common challenges and technical debt that may incur massive ongoing maintenance costs in real-world ML systems. We explore several ML-specific risk factors to account for in system design. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns. We will review many different examples of real-world ML systems and the unique challenges one may encounter to integrate a research ML technique into a production-ready system. We will review unique challenges relevant to each component of an ML system from data collection, feature generation, model learning, model evaluation, model publishing, and acting on the real-world.

In this course, we will also study strategies and principles of distributed ML especially for handling big data in modern production systems. We will learn how to build distributed Deep Learning systems using computer systems best practices. We will study design solutions in ML systems to make them as reliable as a production-ready software system. We will also review design patterns to implement and coordinate ML subsystems. Using powerful frameworks such as Spark, MLlib, Clipper, and Akka, you will learn how to quickly and reliably move from a single machine to a massive cluster. We will then proceed how one can operate a large-scale ML system over time. We will employ the computer systems principles to build ML applications that are responsive, resilient, and elastic. In this course, students will gain hands-on experience applying systems principles to implement scalable learning pipelines. We will also cover various aspects of learning systems, including: automatic differentiation, distributed learning, and scalable model serving. We will finally review best practices of ML at scale in Uber, Spotify, Netflix.

Course Projects and Homeworks

For the course projects/homeworks, students will work on some aspects of distributed ML systems. The details of the projects will become available, but they will be mainly around identifying systems issues in open-source ML systems by looking into GitHub repositories and providing solutions for fixing the issues, building a new model serving system, contributing for a new feature for the following frameworks and engines (essentially something cool either an empirical study or developing a feature):

1. Front-end (Keras, Caffe2), backend engines (Tensorflow, PyTorch, Theano, CNTK, MXNet), and Interoperability tools (Apache ONNX).
2. Modern data architecture patterns for handling big data in ML systems (MapReduce).
3. Configuration and Resource Management, distributed ML engines (Apache Spark), Model Serving Infrastructure (TF serving, Clipper).

If you develop a new feature and submit a pool request to the repositories and that pull request get accepted, your "A" will be guaranteed! See the grading policies for the weights.

Prerequisites

Familiarity with Python and C/C++: we will mainly be using python for the course projects, and C/C++ may be handy for some of the background hacking.

Required Readings

- Jeff Smith, Reactive Machine Learning Systems, MEAP, 2018.
- Deep Learning - The Straight Dope contains useful tutorials and code.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.

Course Policy

I will detail the policy for this course below. Basically, don't cheat and try to learn stuff.

Grading Policy

- **10%** of your grade will be determined by your attendance and participation in class. Generally, ask questions and answer them.
- **60%** of your grade will be determined by the course project(s).
- **30%** of your grade will be determined by coursework throughout the semester.

  The final exam (where you demo your project) will be Tuesday, December 11, at 4:00pm in SWGN 2A24.

Academic Integrity

I would encourage you to discuss or brainstorm with other students or professors, but be aware if you copy/paste from other students/Internet, you will simply fail this course. All the potential Honor Code violations will be reported to the Office of Academic Integrity, which has the authority to implement non-academic penalties as described in STAF 6.25 (http://www.sc.edu/policies/ppm/staf625.pdf).

Disabilities Policy

Any student who has a need for accommodation based on the impact of a documented disability, please contact the Office of Student Disability Services: Phone: 803-777-6142, Email: *sasds@mailbox.sc.edu*, Address: 1523 Greene Street, LeConte College Room 112A, Web: https://www.sc.edu/about/offices_and_divisions/student_disability_resource_center/index.php.

Schedule

Week 01, 09/03 - 09/07: Introduction to production ML systems

Week 02, 09/10 - 09/14: Case study 1 (Uber 1)

Week 03, 09/17 - 09/21: Distributed ML

Week 04, 09/24 - 09/28: Interoperability between ML engines/models

Week 05, 10/01 - 10/05: Case study 2 (Spotify)

Week 06, 10/08 - 10/12: Technical debt in ML

Week 07, 10/15 - 10/19: Systems/Performance issues in ML

Week 08, 10/22 - 10/26: Case study 3 (Netflix)

Week 09, 10/29 - 11/02: Scalability

Week 10, 11/05 - 11/09: Automatic differentiation

Week 11, 11/12 - 11/16: Data architecture patterns

Week 12, 11/19 - 11/23: Project demo

Week 13, 11/26 - 11/30: Model serving

Week 14, 12/03 - 12/07: Case study 4 (Uber 2)

Week 15, 12/10 - 12/14: Showcase final projects