CSCE 585 – Machine Learning Systems

Pooyan Jamshidi

Fall 2019

https://pooyanjamshidi.github.io/mls/ E-mail: pjamshid@cse.sc.edu
Office Hours: TR 13:00 pm – 14:30 pm Class Hours: TR 11:40 am – 12:55 pm
Office: Innovation 2207 Class Room: 300MN 102

Bulletin description

Design, implementation, troubleshooting, testing, and deploying machine learning systems at scale; Deep learning systems stack from design to compiler and hardware deployment; Research directions

Prerequisites

CSCE 240 or 206.

Course Description

When we talk about Machine Learning (ML) or Artificial Intelligence (AI), we typically refer to a technique or an algorithm that gives the computer systems the ability to learn and to reason with data. However, there is a lot more to ML/AI than just implementing an algorithm or a technique. In this course, we will learn the fundamental differences between ML/AI as a technique versus ML/AI as a system in production. A ML/AI 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/AI systems responsive, resilient, and elastic. ML/AI systems are different than other computer systems when it comes to building, testing, deploying, delivering, and evolving. ML/AI 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/AI systems (e.g., performance issues, memory leaking, communication issues, multi-GPU issues, etc). The focus of this course will be primarily on *deep learning systems*, but the principles will remain similar across all ML/AI systems. More specifically, we will mainly target the following concerns:

- Data scientists often make great progress at building models with cutting edge techniques but turning those models into products is challenging. For example, data scientists may work with unversioned notebooks on static data sets and focus on prediction accuracy while ignoring scalability, robustness, update latency, or operating cost.
- Software engineers are trained with clear specifications and tend to focus on code, but may not be aware of the difficulties of working with data and unreliable models. They have a large toolset for decision making and quality assurance but it is not obvious how to apply those to intelligent systems and their challenges. To what degree can existing SE practices be used for building intelligent systems? To what degree are new practices needed?

This course adopts a systems perspective on building intelligent systems, focusing on what a data scientist can do to turn a machine learning idea into a scalable and reliable product. We will also cover some core machine learning topics such as:

- Machine Learning Systems: Concepts, Challenges, and Solutions
- Optimization, Neural Nets, and Learning Theory
- Deep Convolutional Neural Networks
- Deep Learning System Stack
- Backprop and Automatic Differentiation
- Hardware Backends: GPU, CPU, TPU
- Hardware and Memory Optimization
- Machine Learning Platforms and Model Serving
- Distributed and Scalable Machine Learning
- Troubleshooting Deep Neural Networks
- Machine Learning System Testing and Deployment at Scale
- Setting up Machine Learning Projects and Teams
- Highly-Configurable Machine Learning Systems and Transfer Learning
- Research Directions

Learning Outcomes

1. Explaining differences between ML as predictive technique and as a computer system.
2. Describing how a distributed ML system works in production and insight into challenges.
3. Locating technical debt in building ML systems.
4. Employing design strategies and best practices to mitigate technical debt.
5. Incorporating ML-based components into a larger system.
6. Stating the principles that govern ML systems.
7. Building systems that are more capable, both as software and as predictive systems.
8. Identifying systems issues and apply strategies to avoid them in production ML systems.
9. Picking the right framework and compute infrastructure
10. Troubleshooting training and ensuring reproducibility
11. Deploying the model at scale

Why this course is required in modern computer science curriculum?

In this course, we will understand central principles of **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 different aspects of ML systems. For more details, please refer to the following page: https://pooyanjamshidi.github.io/mls/projects/. There are materials for previous editions of this course.

Readings and Textbooks

No textbook is required, here are some recommended materials for reading:

- Hulten, Geoff. Building Intelligent Systems: A Guide to Machine Learning Engineering. Apress; 1st ed. edition (March 7, 2018). http://intelligentsystem.io/book/
- Deep Learning Book (https://deeplearningbook.org) is a comprehensive material for the field.
- CS231n (http://cs231n.stanford.edu/) is a great course to get started on DL for vision.
- Uber Engineering: https://eng.uber.com/
- The Netflix Tech Blog: https://medium.com/netflix-techblog
- Spotify Labs: https://labs.spotify.com/

Course Policy

Communication

We envision several routes of communication for this course:

- The main mode of electronic communication between students and TAs, as well as amongst students, will be through Piazza. It is intended for general questions about the course, clarifications about assignments, student questions to each other, discussions about material, and so on.
- If you need to contact the instructor, you should email with the provided address.

Enrollment

All CSE majors can enroll in this course. The same for other majors, but they have to wait to sign up after the CSE majors, and might have to fill out the override request form: https://cse.sc.edu/undergraduate/forms/override-request.

Academic Integrity

I would encourage you to discuss or brainstorm with other students or professors about the assignments, projects and homeworks, but submissions should acknowledge all collaborators and sources consulted. We will actively check for code and other kinds of plagiarism (both from current classmates and other available online sources). We trust you all to submit your own work, and you are expected to practice the highest possible standards of academic integrity. Any deviation from this expectation will result in a minimum academic penalty of your failing the assignment, and will result in additional disciplinary measures including referring you to the Office of Academic Integrity. Violations of the University's Honor Code include, but are not limited to, plagiarism, cheating, falsification, complicity, and any other form of academic misrepresentation. For more information, see https://www.sa.sc.edu/academicintegrity/.

Disabilities Policy

Reasonable accommodations are available for students with a documented disability. If you have a disability and may need accommodations to fully participate in this class, contact the Office of Student Disability Services: 777-6142, TDD 777-6744, email sasds@mailbox.sc.edu, or stop by LeConte College Room 112A. All accommodations must be approved through the Office of Student Disability Services.

Late Work

Homework assignments are due at the time listed on Dropbox. No late work will be accepted. Submitting all assignments is a necessary condition for passing this class.

Syllabus Change Policy

This syllabus is a guide and every attempt is made to provide an accurate overview of the course. However, circumstances and events may make it necessary for the instructor to modify the syllabus during the semester and may depend, in part, on the progress, needs, and experiences of the students. Changes to the syllabus will be made with advance notice.

Policies and Procedures

This section contains some general rules that will be enforced during this course. Please review these guidelines carefully. The course is governed by the policies and procedures of the university (http://www.sc.edu/policies/ppm/staf625.pdf). Violations of this code can result in actions varying from a failing grade to expulsion from the university.

Grading Policy (undergraduate students only)

- **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. There will be only one project per team of students which will be formed in the first 3 weeks. Each team will be working on their own project throughout the semester and they will present/demo their work at the end of semester. The grade will be per student depending on her/his contribution to the project, progress throughout the semester, and the quality of the final presentation.
- **30%** of your grade will be determined by homeworks throughout the semester. Homeworks are assignments that need to be delivered in 1-2 weeks, so students do these assignments regularly throughout the semester.

Grading Policy (graduate students only)

- **10%** of your grade will be determined by your attendance and participation in class. Generally, ask questions and answer them.
- **40%** of your grade will be determined by the course project. There will be only one project per team of students which will be formed in the first 3 weeks. Each team will be working on their own project throughout the semester and they will present/demo their work at the end of semester. The grade will be per student depending on her/his contribution to the project, progress throughout the semester, and the quality of the final presentation.
- **20%** of your grade will be determined by the project report. In addition to writing code and presenting the results of projects, as I expect for all students, graduate students are required to write a report, see details here: `https://pooyanjamshidi.github.io/mls/projects/`.
- **30%** of your grade will be determined by homeworks throughout the semester. Homeworks are assignments that need to be delivered in 1-2 weeks, so students do these assignments regularly throughout the semester.

Grades are on the following fixed scale:

A	[90 – 100]
B+	[86 – 90)
B	[75 – 86)
C+	[70 – 75)
C	[60 – 70)
D+	[55 – 60)
D	[40 – 55)
F	[0 – 40)

Course Schedule

https://pooyanjamshidi.github.io/mls/lectures/

- Week 1: Machine Learning Systems: Concepts, Challenges, and Solutions
- Week 2: Optimization, Neural Nets, and Learning Theory
- Week 3: Deep Convolutional Neural Networks
- Week 4: Deep Learning System Stack
- Week 5: Backprop and Automatic Differentiation
- Week 6: Hardware Backends: GPU, CPU, TPU
- Week 7: Hardware and Memory Optimization
- Week 8: Machine Learning Platforms and Model Serving
- Week 9: Distributed and Scalable Machine Learning
- Week 10: Troubleshooting Deep Neural Networks
- Week 11: Machine Learning System Testing and Deployment at Scale
- Week 12: Setting up Machine Learning Projects and Teams
- Week 13: Highly-Configurable Machine Learning Systems and Transfer Learning
- Week 14: Research Directions