CSCE 585 – Machine Learning Systems

Pooyan Jamshidi

Fall 2020

Website: **https://pooyanjamshidi.github.io/mls/** E-mail: `pjamshid@cse.sc.edu`
Office Hours: by appointment Class Hours: Async & Sync
Office: Virtual (Zoom) Class Room: Blackboard Ultra

Bulletin description

Design, implementation, testing, and deploying machine learning systems at scale; Deep learning systems stack from design to computer system and hardware; Research directions (Robust Optimization, Causal AI, AI/ML Systems in Space, Adversarial ML, AI Systems for Social Good, AI Systems for Diversity&Inclusion, Safe AI, Transfer Learning, Deep Reinforcement Learning, Neuromorphic Computing).

Prerequisites

CSCE 240 (Advanced Programming Techniques) or CSCE 206 (Scientific Applications Programming).

Course Description

When we talk about Artificial Intelligence (AI) or Machine Learning (ML), we typically refer to a technique, a model, 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 than just implementing an algorithm or a technique. In this course, we will learn the fundamental differences between AI/ML as a model versus AI/ML as a system in production. An ML system[1] 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. ML systems are different from other computer systems when it comes to building, testing, deploying, delivering, and evolving. Machine learning systems also have unique challenges when we need to change the architecture or behavior of the system. Therefore, it is essential to

[1] We use AI Systems and ML Systems interchangeably

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 leaks, 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 systems.

More specifically, this course is designed for the following audience:

- **Computer Scientists, Data Scientists, AI/ML/Systems/Math/Statistics Researchers** 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 operational costs.
- **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.
- **Scientists, Researchers, Engineers, Lawyers, Medical Experts, Physicians, Psychologists, Business Experts, and many other Professionals** have heard about AI/ML and would like to know more about the details and how to start or use AI/ML in their profession or their research. It is important to learn how to design ML systems in a principled and systematic way that satisfies certain properties like safety, bias, etc. If we do not learn about these important principles, we may have many ML systems that undermine our human values in our society.

Learning Outcomes

1. Learning to design ML Systems to solve practical problems.
2. Explaining differences between ML as a model and as a system deployed at scale.
3. Describing how an ML system works in production and insights about challenges (AIOps).
4. Locating technical debt in building ML systems.
5. Employing design strategies (such as concepts) and best practices to mitigate technical debt.
6. Incorporating ML-based components into a larger system (e.g., Cyber-Physical Systems).
7. Stating the underlying principles that govern ML systems.
8. Building systems that are more capable, both as software and as predictive systems.
9. Identifying systems faults and apply strategies to identify root causes in ML systems.
10. Picking the right framework and compute infrastructure and trade-off space.
11. Understanding performance landscape (energy, inference time) and optimization.
12. Troubleshooting training and ensuring the reproducibility of results.
13. Deploying predictive models on resource-constrained environments (e.g., NVIDIA Xavier)

Topics

- Machine Learning Systems: Concepts, Challenges, and Solutions
- Machine Learning Systems and Software Stack
- Optimization, Neural Nets, and Learning Theory
- Backprop and Automatic Differentiation
- Hardware for Machine Learning Systems: GPU, CPU, TPU, Neuromorphic Computing
- Machine Learning Accelerators (ML Compilers)
- In-memory Computation: Memory Locality and Memory Bandwidth
- Quantized and Low-precision Machine Learning
- Deployment and Low-latency Inference: Platforms and Model Serving
- Distributed and Scalable Machine Learning
- Machine Learning System Testing and Debugging at Scale
- Setting up Machine Learning Projects and Teams
- Research Directions: Robust Optimization, Causal AI, AI/ML Systems in Space, Adversarial ML, AI Systems for Social Good, AI Systems for Diversity&Inclusion, Safe AI, Transfer Learning, Deep Reinforcement Learning.

Why this course is important and should be required in the modern computer science curriculum?

In this course, we will understand the 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 with how one can operate a large-scale ML system over time. We will employ 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. And most importantly, this course will prepare students to take on jobs in companies that massively use ML such as Google, Facebook, Uber, and similar types. This course is also very much aligned for those who want to start doing research in some of the fascinating AI/ML topics we will cover in this course. Overall, I am hopeful that this course will have an impact (even very small) for our society, humanity, our future, and beyond.

Course Projects

CSCE 585 is a project-based course via which you learn how to build an ML System with certain characteristics. We will define a couple of project description, so students can choose to work on one of the projects related to ML Systems. There are materials for previous editions of this course on the course website, we will update this page soon with links to the project descriptions for this edition. For more details, please refer to the following page: https://pooyanjamshidi.github.io/mls/projects/. Announcements will be sent via Piazza once the project descriptions are available.

Readings and Textbooks (Optional)

No textbook is required, here are some recommended materials for reading (all readings/materials comply with copyright/fair use policies):

- 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.

Course Policy

Communication

We envision several routes of communication for this course:

- **Student-to-Instructor & Student-to-Student:** The only mode of communication between students, instructor, and TA will be all electronically through the following tools/services:
  - **Blackboard Collaborate Ultra**: All sync lectures will be through Blackboard Collaborate Ultra (they will be also recorded) at the same time the class is scheduled. Note that we will not have sync lectures all weeks. We will only have the sync lectures whenever is needed for question answering, review, discussions, project presentations, and guest lectures and these will be communicated via Piazza. I will pre-record some short lectures for most of the content in the syllabus.
  - **Piazza**: It is intended for general questions about the course, clarifications about assignments, questions about research, discussions about the material, and so on. All students *must* enroll in Piazza (the link can be found on the course website).
  - **GitHub**: We will use GitHub for all projects and their deliverable. We will create repositories to share the specification of each project and you should commit your code there. We will only look at it when you want us to do so and for the final evaluation. Note that all project deliverables should be a push to the repositories for final evaluation. Please do not email any project deliverable to us, they will be simply ignored. Learning GitHub is easy and the TA can show how to use it.
  - **Gradescope**: We will use GradeScope for all assignments. All students are required to make an account at Gradescope and enroll using this code that will be provided on the course website.
  - **Slack**: I will also set up a workspace for the course.
  - **Email**: I would strongly encourage you to ask your question via Piazza so others can also benefit from the questions and learn from the answers. However, if you need to contact the instructor for something that cannot be shared with others (personal issues or something confidential), you can email me or the TA with the provided address. Before sending an email, please think whether it is really necessary or can be shared on Piazza (it is great and you must use it regularly throughout the course).
  - **Zoom**: The instructor and TA will use Zoom for one-2-one meetings or the project team if needed. These are all by appointment.
- **Student-to-Content:** All course materials (e.g., lectures, notes, project descriptions, templates, assignments, syllabus, policies) can be accessed via this link: https://pooyanjamshidi.github.io/mls/. Note that I will update these materials throughout the semester, so to find out about the latest versions please regularly visit this link. I will announce any update of content via Piazza too.

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 lecturers/TA about the assignments and projects, 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 Student Disability Resource Center: 803-777-6142, TDD 803-777-6744, email sadrc@mailbox.sc.edu, or stop by LeConte College Room 112A. All accommodations must be approved through the Student Disability Resource Center. See https://www.sa.sc.edu/sds/.

Late Work

All assignments and projects have due dates. No late work will be accepted. Submitting all assignments and the project 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 will be reported.

Grading Policy (undergraduate students only)

- **Course Project**: 70% of your grade will be determined by the course project. There will be only one project per team of students (we will discuss how to form a team in the first lecture), which must be formed in the first 3 weeks. I will provide details about the project specification int he first 3 weeks. Students will present their progress throughout the semester. The grade will be per student depending on her/his contribution to the project, regular progress throughout the semester, and the quality of the final presentation, results, and delivered prototype.
    - Optional (encouraged): In addition to writing code and presenting the results of the projects, undergraduate students may write a report (that can be submitted as a workshop paper), see details here: https://pooyanjamshidi.github.io/mls/projects/. We will discuss more details about the projects in early lectures.
- **Assignments**: 30% of your grade will be determined by assignments throughout the semester. Assignments will be very much aligned to the projects and they need to be delivered in 1-2 weeks, so students do these assignments regularly throughout the semester.

Grading Policy (graduate students only)

- **Course Project**: 70% of your grade will be determined by the course project. There will be only one project per team of students (we will discuss how to form a team in the first lecture), which must be formed in the first 3 weeks. I will provide details about the project specification int he first 3 weeks. Students will present their progress throughout the semester. The grade will be per student depending on her/his contribution to the project, regular progress throughout the semester, and the quality of the final presentation, results, and delivered prototype. In addition to writing code and presenting the results of the projects, as I expect for all students, *graduate students* are required to write a report (that can be submitted as a workshop paper), see details here: https://pooyanjamshidi.github.io/mls/projects/. We will discuss more details about the projects in early lectures.
- **Assignments**: 30% of your grade will be determined by assignments throughout the semester. Assignments will be very much aligned to the projects and they 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

We will have some very exciting **guest lectures** from very known researchers in the areas of AI/ML/Systems on the following topics: Robust Optimization, Causal AI, AI/ML Systems in Space, Adversarial ML, AI Systems for Social Good, AI Systems for Diversity&Inclusion, Safe AI, Transfer Learning, Deep Reinforcement Learning, Neuromorphic Computing. The exact schedule for guest lectures is tentative and is subject to change. For details about the lectures, please refer to (will be updated throughout the semester): https://pooyanjamshidi.github.io/mls/lectures/

- **Week 1:** Machine Learning Systems Design: Concepts, Challenges, and Solutions
- **Week 2:** Machine Learning Systems and Software Stack
- **Week 3:** Optimization, Neural Nets, and Learning Theory
- **Week 4:** Backprop and Automatic Differentiation
- **Week 5:** Research Directions (Guest Lectures)
- **Week 6:** Hardware for Machine Learning Systems: GPU, CPU, TPU, Neuromorphic
- **Week 7:** Machine Learning Accelerators (ML Compilers)
- **Week 8:** In-memory Computation: Memory Locality and Memory Bandwidth
- **Week 9:** Research Directions (Guest Lectures)
- **Week 10:** Quantized and Low-precision Machine Learning
- **Week 11:** Deployment and Low-latency Inference: Platforms and Model Serving
- **Week 12:** Distributed and Scalable Machine Learning
- **Week 13:** Machine Learning System Testing and Debugging at Scale
- **Week 14:** Setting up Machine Learning Projects and Teams
- **Week 15:** Research Directions (Guest Lectures)