



CS 59200: ML System Co-Design

Algorithm–Hardware Co-Design for Machine Learning Systems

Instructor: Haoran You, <haoran@purdue.edu>

Semester: Fall 2026

Instructional Modality: In-person seminar meetings (LWSN B134)

Meeting Day(s) and Time(s): TR / 1630

Office Hours: Tuesdays 3:00–4:00 PM, DSAI 2123 (subject to change)

Course Description

Modern machine learning models such as large language models and diffusion models require massive computational resources. Achieving efficient training and serving requires tight co-design across algorithms, systems, and hardware accelerators.

In this course, we will explore the algorithm–hardware co-design principles for modern ML systems, covering how model compression techniques, system optimizations, and hardware accelerators interact to determine performance, efficiency, and scalability. We will focus on papers from MLSys, ISCA, MICRO, HPCA, ASPLOS, OSDI, SOSP, DAC, or related venues, where top machine learning systems and computer architecture research is published. Students will learn how ML workloads are mapped to modern accelerators (e.g., GPUs, TPUs, and other AI accelerators), how algorithms are co-designed with hardware accelerators, and how algorithmic innovations can reduce computation and memory costs while making acceleration easier for hardware.

Prerequisites

This course requires a background in Computer Science and knowledge of basic machine learning, computer architecture or systems, Python, and deep learning frameworks (e.g., PyTorch, TensorFlow, CUDA). Graduate students with relevant backgrounds are welcome to register. Undergraduate students with the required knowledge, i.e., a grade of B or better in CS 25000, CS 25100, and CS 25200, may also register.

Grading Criteria

- Paper Presentation: 40%
- Participation: 10%
- Midterm: 20%
- Final Project: 30%

This is a seminar course structured around student presentations. We will read and discuss research papers during seminar sessions, and the paper list will be posted along with the class schedule. The course project includes reproducing a relevant algorithm, system, or computer architecture co-design paper, such as implementing the algorithm or reproducing the paper's results using hardware implementation or simulation, or designing improved ML systems based on existing open-source projects.

Learning Outcomes

By the end of this course, students will be able to:

- Understand performance bottlenecks in modern ML workloads.
- Analyze hardware–software interaction in ML systems.
- Design algorithmic optimizations that improve hardware efficiency.
- Implement system-level optimizations for ML workloads.
- Conduct research on efficient ML systems.