Syllabus for CS395T: Foundations of Machine Learning for Systems Researchers

Course Overview

Rapid advances in machine learning have enabled us to build systems with capabilities that were unimaginable just a few years ago, such as AlphaGo, DeepSeekCoder, and AlphaEvolve. The goal of this course is to analyze the key breakthroughs that underlie these kinds of systems, and to understand how they can be used to build systems for solving other problems.

This semester, the course will focus on the generation of high-performance kernels using AI/ML techniques. This is a more challenging problem than automatically generating code for websites, for example, since there is not much training data available, particularly for the many new and innovative processors that have appeared recently.

Lectures will cover three main machine learning technologies: deep neural networks, reinforcement learning, and evolutionary computing. Unlike standard machine learning courses, this material is presented using PL/systems concepts such as dataflow analysis. Lectures will also discuss how high-performance kernels are implemented manually on CPUs and GPUs, using important kernels like FlashAttention as examples. Students will present recent papers in conferences like NeurIPS, ICML, ICLR for automating kernel generation.

Prerequisites

- Proficiency in Python
- Familiarity with parallel programming at the level of CS 377P
- Familiarity with calculus, statistics, linear algebra, and strong mathematical skills
- Coursework or equivalent experience in AI/ML at the level of CS 342 is strongly recommended

Course Outcomes

Build a solid foundation in DNNs, RL, and evolutionary computing—including limitations. Understand how these technologies are deployed in modern systems. Understand how to write high-performance kernels manually for modern processors with CPUs, GPUs, and memory hierarchies. Evaluate and leverage these technologies to build systems that can automatically generate high-performance kernels for CPUs and GPUs.

Coursework

- Paper presentations (30%)
- Paper reviews (10%)
- A substantial term project (50% total)

  - Project Proposal (10%)
  - Final Project Paper (25%)
  - Project Presentation (15%)
- Participation and reading quizzes (10%)

Academic Honesty

You may discuss concepts with classmates, but all written work and programming assignments must be your own or your project team's work when teamwork is permitted. You may not search online for existing implementations of algorithms related to the programming assignments, even as a reference. Students caught cheating will automatically fail the course and will be reported to the university. If in doubt about the ethics of any particular action, talk to the instructor or the TA.

Notice about students with disabilities

The University of Texas at Austin provides upon request appropriate academic accommodations for qualified students with disabilities. For more information, contact the Division of Diversity and Community Engagement — Services for Students with Disabilities at 512-471-6529.

Course Staff

Dr. Keshav Pingali
Email: pingali@cs.utexas.edu
Office: POB 4.126
OH: Fri 1–2 PM

TA: Soumyabrata Chaudhuri
Email: sc74532@my.utexas.edu