Online Research Seminar Syllabus

1. Overview

Title	Hardware for Machine Learning		
Mode	Leading Instructor Sessions & Teaching Fellow Sessions		
Prerequisites	**High School Students**	Required course/Knowledge	Basic understanding of system principles and design: data structures, operating systems, cache coherence, processor programing. C/C++/python coding knowledge. Some understanding of ML is useful but not mandatory.
		Recommended Materials for preparing for the course	MIT Open Courseware 6-004-computation-structures
	College Students	Required course/Knowledge	Basic understanding of system principles and design: data structures, operating systems, cache coherence, processor programing. C/C++/python coding knowledge. Some understanding of ML is useful but not mandatory.
		Recommended Materials for preparing for the course	MIT Open Courseware 6-004-computation-structures

2. Program Introduction and Objectives

Course Description	The course will guide students through the world of machine learning hardware and how systems architects are able to consciously provide tremendous computation power advancements to keep up with the demand of machine learning and artificial intelligence requirements. The course will cover the fundamental of GPU architectures and programming and industry hardware deigns such as Google's Tensor Processing Units (TPUs) that power Google's cloud.
Software/Tools (if any)	**Tensorflow**

3. Program Schedule

Week	Leading Instructor Session	Teaching Fellow Session (lab/case study, etc.)	Assignment	Reading Materials

1	Topic	**Overview of Machine learning hardware and systems.**			
	Detail	The lecture will cover the fundamentals of hardware and systems required to drive machine learning today. The lecture will provide fundamental knowledge to topics around cloud deployments of machine learning, their system and hardware requirements.			
2	Topic	**Machine Learning Systems**		**Assignment on machine learning systems based on Tensorflow.** **Deadline on Lecture 3.**	Tensorflow
	Detail	The lecture will cover the fundamentals behind machine learning systems such as Tensorflow and how system designs interacts with the requirements of machine learning.	The teaching fellow should setup a lab that will guide students through Tensorflow.		
3	Topic	**GPUs for Machine Learning**		**Assignment on GPU CUDA for ML.** **Deadline on Lecture 4.**	GPU CUDA programming.
	Detail	The lecture will cover fundamental GPU architecture characteristics and how GPUs are used in the cloud for machine learning.	The teaching fellow should guide students through GPU CUDA programming basics.		
4	Topic	**Accelerator Hardware Design for Machine Learning**			Google TPU
	Detail	The lecture will cover machine learning hardware architectures with a special focus on Google's Tensor Processing Unit (TPU).			
5	Topic	**Research Workshop Group 1: Continued discussion of research workshop (remaining students/groups).**	**Research Workshop Group 1: Continued discussion of research workshop (remaining students/groups).**		
	Detail	**See section 5.**	Fellow should prepare a poster template for students to use.		
6	Topic	**Research Workshop Group 2: Continued discussion of research workshop (remaining students/groups).**	**Research Workshop Group 2: Continued discussion of research workshop (remaining students/groups).**		
	Detail	**See section 5.**	Fellow should prepare a poster template for students to use.		

7	**Final Oral Presentation and Written Reporting**

4. Problem Sets/Written Assignments/Quizzes

Total Number of Assignments	_2__times
Submission Deadline	1 weeks after lecture.
Will there be Quizzes? How often/how many?	No
Other Requirements (if any)	

5. Final Oral and Written Project

Detailed requirements of the final project:

Projects will cover all the lectures material. Projects should be completed by teams of 3-5 students.

- The project will require the development of a Tensorflow ML serving server. The goal of the students is to develop a machine learning server that will utilize multiple CPU cores and GPUs (if available). Clients will send ML inference requests to the server and get ML inference responses. Each server should support multiple ML models such as Resnet and BERT. The performance of the server will be measured in terms of requests per second, number of clients, and the number of cores utilized. The evaluation should compare the impact of the batching size and tail latency.

5.1 Final Oral Presentation

- Oral Project Requirements (e.g: if slides needed; Format; Criteria; Deadline):

Poster presentation that focuses the developed project. The poster should include an introduction section, a design section, and an evaluation section.

Due at poster presentation day. Each group students (usually 3-5 students) will need to develop a poster that explains their work in detail. It should focus on the highlights of their implementation and put emphasis on their evaluation results.

5.2 Will you require a written final report as well?

- Written Project Requirements (e.g: word count; style; criteria; Deadline):

Two-page summary of project work with evaluation results.
Deadline at the end of the class. The summary should include, the design of the server architecture, the ML models used for experimentation, a detailed project implementation section and a thorough discussion of the results including figures that show the scalability of the design and its performance. The format of the report should be double-column conference style report with a font of 11pt in Times and 1-inch page margins.

6. Suggested Future Research Fields/Direction/Topics

Students that aim to pursue a career in ML systems should focus on learning more about the implications of ML in system design and architecture. This course will provide the basics for students to dive deeper into ML systems for either industry or PhD positions. Sharpening their research skills enthusiastic students should focus on gaining a better understanding of current research into ML systems architectures.

7. Instructor Introduction

Instructor Title
Prof. Dimitrios

Instructor Bio
Dimitrios is an assistant professor in the Computer Science Department at Carnegie Mellon University. His research bridges computer architecture and operating systems with a focus on performance, security, and scalability. He has received several awards for his cross-cutting research including the NSF CAREER award, four Meta Faculty Awards in systems and security, the joint 2021 ACM SIGARCH & IEEE CS TCCA Outstanding Dissertation award for “contributions to redesigning the abstractions and interfaces that connect hardware and operating systems”, the David J. Kuck Outstanding Ph.D. Thesis Award for the best PhD thesis in the computer science department at the University of Illinois at Urbana-Champaign, an ISCA Best Paper Award, two ASPLOS Best Paper Awards, and three IEEE MICRO Top Picks.

Instructor Profile Photo