Topics in Machine Learning: Large Models

Important Links

- **Course web site:** `https://www.cs.toronto.edu/~cmaddis/courses/csc2541_w25/`
- **Quercus:** `https://q.utoronto.ca/courses/379872`
- **MarkUs:** `https://markus.teach.cs.toronto.edu/markus/main/login_remote_auth`

Course materials (schedule, slides, readings, assignments) can be found on the course web site.

Overview

Large language models have revolutionized artificial intelligence and machine learning. These models, trained on massive datasets, can generate human-like text, code, and (apparently) engage in complex reasoning tasks. At the core of these breakthroughs are so-called empirical scaling laws that show how model capabilities emerge predictably with increased model size and data size. This predictability has motivated an immense industrial effort to build and deploy very large models.

The course will focus on understanding the practical aspects of large model training through an in-depth study of the Llama 3 technical report (Grattafiori et al, 2024). We will cover the whole pipeline, from pre-training and post-training to evaluation and deployment.

Students will be expected to present a paper, prepare code notebooks, and complete a final project on a topic of their choice. While the readings are largely applied or methodological, theoretically-minded students are welcome to focus their project on a theoretical topic related to large models.

Prerequisites

This is a graduate course designed to guide students in an exploration of the current state of the art. So, while there are no formal prerequisites, the course does assume a certain level of familiarity with machine learning and deep learning concepts. A previous course in machine learning such as CSC311 or STA314 or ECE421 is required to take full advantage of the course, and, ideally, students will have taken a course in deep learning such as CSC413. In addition, it is strongly recommended that students have a strong background in linear algebra, multivariate calculus, probability, and computer programming.

Teaching Staff

Instructor: Chris J. Maddison
Teaching Assistants: Ayoub El Hanchi, Frieda Rong
Staff e-mail: `csc2541-large-models@cs.toronto.edu`

Please send course-related e-mails to the instructor/staff emails above. If you have a private matter that you would like to discuss with the instructor, you can email `cmaddis@cs.toronto.edu`.

Schedule

Lecture Times

- LEC0101: Fridays 11:00AM-1:00PM.

Lecture Structure Every week, we will read a section of the Llama 3 technical report or some additional "core" readings. In weeks 1-2, we will deliver lectures that cover an overview of the course and some of the key concepts in machine learning and deep learning. In weeks 3-12, students will deliver presentations that augment and expand on that week's Llama 3 report section. Every student will present a paper once during the course.

We recommend that you read at least the core readings and as many of the additional papers as possible. We won't check whether you've read the assigned readings, but you will get more out of the course if you do. See the course web page for information about topics.

Recordings

Lecture recordings will be generated and posted automatically on the OCCS Student App.

This course, including your participation, will be recorded on video and will be available to students in the course for viewing remotely and after each session.

Course videos and materials belong to your instructor, the University, and/or other sources depending on the specific facts of each situation, and are protected by copyright. Do not download, copy, or share any course or student materials or videos without the explicit permission of the instructor.

For questions about recording and use of videos in which you appear please contact your instructor.

Course Evaluation

- 25% – Paper presentation and code notebook (due at the start of your presentation)
- 15% – Project proposal (due Feb 14)
- 60% – Final project (due April 9)

Details will be posted on the course web site.

Submission Policies

There are 3 assignments to submit: presentation slides and a code notebook, a project proposal, and a final project report.

Format. The presentation slides, project proposal, and final project write-up must be submitted in PDF format through MarkUs. We encourage typesetting using LaTeX, but other formats are acceptable as long as they are legible.

The code notebook must be written in Python and submitted through MarkUs as a Python notebook (`.ipynb` file format) that can be run on Google Colab (`https://colab.research.google.com`). Your implementation can use any Python package as long as the notebook is self-contained and runnable on Google Colab by us. We will not attempt to decipher a buggy notebook or solve broken dependencies.

Lateness. Assignments will be accepted up to 3 days late, but 10% will be deducted for each day late, rounded up to the nearest day. No credit will be given for assignments submitted after

3 days. Extensions will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the instructor at least one week before the due date.

Collaboration policy. Collaboration on the presentation and final project is allowed. The teams do not need to be the same for the presentation and the final project. The report for the final project should list the contributions of each team member.

Remarks. Remark requests will be considered by the same TA who marked the assignment. The deadline for requesting a remark is one week after the marked assignments are returned. Remarks may result in a decrease in the grade.

Policy on The Use of Artificial Intelligence

Large-language-model-based chatbots like ChatGPT and Claude can be very useful educationally, and you're encouraged to take advantage of them. The only restriction is that **you are not allowed to use them in a way that trivializes an assignment**, such as asking them how to solve a problem.

Along with each assignment, **you must submit (on MarkUs) any chat transcripts directly related to the assignment**. We'll err on the side of permissiveness (i.e. you won't be penalized as long as you're acting in good faith), but we may revise the policy if we see that use of chatbots is reducing the educational value of the assignments. If a chatbot substantially more powerful than the current frontier models is released during the term, we may amend the policy to address that.

Computing Support

As a graduate-level course, we are assuming that most students in this class have a laboratory affiliation which grants them access to computing resources which they may use for the purposes of this class. If you do not have access to compute via a laboratory affiliation, please consider using Google Colab (`https://research.google.com/colaboratory/`) to run your experiments, or contact the teaching staff **as early as possible** and we will do our best to provide some for you.

Accessibility Support

We are committed to making our classroom an accessible environment. If you require additional academic accommodations, please contact UofT Accessibility Services as soon as possible, `studentlife.utoronto.ca/as`, and contact the course staff.

Auditing Policy

It is possible for non-enrolled persons to audit this course (sit in on the lectures) *only if the auditor is a student at U of T, and no University resources are to be committed to the auditor*. This means that students of other universities, employees of outside organizations, or any other non-students, are not permitted to be auditors.