Statistics 8931, Fall 2024

Generative AI: Principles and Practices

Course Instructor: Prof. Jie Ding
Schedule: every Friday from 9:05 am to 11:35 am, 9/6/2024 - 12/6/2024
Location: Ford Hall 110
Office Hours: TBD or by email appointment

Online Streaming (for those who cannot make it in-person): Zoom
One-time registration for Zoom access:
https://umn.zoom.us/meeting/register/tJcrdeqrrTstEtxMvleVLgI8YO80fIEuwu1x
After registering, you will receive a confirmation email containing information about joining the meeting.

Course Design Team: Xun Xian, Ganghua Wang, Jin Du, Qi Le, Xinran Wang, An Luo

Lab Instructor/Teaching Assistant (TA): None

Course Purpose: To establish a solid foundation in Generative AI, promoting both critical and creative thinking

Course Audience: Open to PhD students from all disciplines interested in AI, from novices to experienced researchers

Prerequisites: Basic calculus, algebra, and probability, (preferred) introductory-level machine learning

Computing Requirements:
Proficiency in Python, as the course involves extensive numerical studies
Familiarity with Markdown and LaTeX for interacting with Github repo
Educational GPU-cloud accounts will be provided for free to all students

Textbook and References:
This course will not require traditional textbooks. Instead, Prof. Jie Ding will curate a comprehensive collection of course materials from scratch, including lecture notes, research references, sample codes, and related toolsets. These resources will be made available through a GitHub-based open-source course website and other course-specific tools. We encourage contributions from all students to ensure the content remains balanced and up to date with the latest advancements in the field.

- Public Resources: We will heavily utilize the GitHub platform for organizing course materials, sharing projects, and submitting homeworks. We will sometimes use Huggingface to access, host, and release pre-trained models, engage in community projects, and demonstrate application products.

- Optional Reference Text: Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville is recommended for students seeking a basic understanding of deep learning that underpin many generative AI models.

Communication Channels:
Each lecture in this course is designed to be highly interactive through a blend of instructive presentations and group-based exercises. You can engage directly with peers and the instructor during sessions. For matters that cannot be addressed during class, please utilize the appropriate communication channels as outlined below:

- **Self-learning and discussions**: For queries about course material, open discussions, and homework, use our course-specific Slack named "GenAI Course, Fall 2024."

- **Code-related Q&A**: If you have issues or new findings to share related to course-related coding, post your queries in the GitHub Issues section of our course-specific repository.

- **Homework submission**: All homework should be submitted through your designated student-specific GitHub repository.

- **Research ideas to discuss**: For in-depth discussions about research ideas or complex issues, attend the office hours or schedule an appointment with the course design team.

- **Logistics, disabilities, and emergent medical situations**: Please email the instructor privately at dingj@umn.edu with the subject line "Stat 8931 GenAI Fall 2024".

Course Coverage Plan:

Overview
This course is thoughtfully created to provide comprehensive coverage of Generative AI, merging foundational principles with cutting-edge applications. The course will delve into a variety of topics, ensuring that students grasp both the theoretical principles, algorithmic underpinnings, and practical implementations. While lectures will emphasize key areas, diligent engagement with assigned readings is expected for a holistic understanding.

Course Content

Chapter 1. Quick Study of Deep Learning

Dive into the essentials of deep learning, exploring core model architectures such as CNNs, VAEs, and ResNets. Understand the computational underpinnings, including frameworks like Pytorch and Tensorflow, optimization methods like SGD and ADAM, and foundational principles. We'll also showcase diverse applications, spanning natural language processing to computer vision.

Chapter 2. Large Language Modeling

Language modeling's trajectory from N-gram models through Word2vec to contemporary pre-trained models will be charted. We will dissect the probabilistic methodologies that underlie language modeling, including various decoding algorithms and their inherent constraints. We will also dissect the transformer architecture to understand its core components, including attention mechanisms, positional encoding, and the significance of self-attention over recurrent or convolutional layers. We will then examine the myriads of transformer variants that have emerged and evaluate the empirical observations that have been made about transformer models, such as their surprising ability to perform zero-shot learning, their emergent linguistic abilities at scale, and their application in few-shot learning scenarios.

Chapter 3. Training LLMs from Scratch

This lecture will introduce standard steps and techniques used in training LLMs from scratch, addressing tokenizer training, self-supervised pre-training, and instruction finetuning. The lecture will also connect the algorithms and fundamental principles in optimization and statistical learning.

Chapter 4. Human-Value Alignment

Explore the critical aspect of human-value alignment in AI development, ensuring that AI systems act in accordance with human values and ethical principles. This involves exploring various algorithmic approaches such as Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO), both of which are designed to fine-tune AI behavior to match human expectations and preferences. We will discuss the ethical and practical implications of AI decisions and actions, emphasizing the importance of aligning AI systems with societal norms and individual values. The chapter covers recent advancements in the field, including innovative techniques for integrating human feedback into the training process, methods for quantifying and evaluating alignment, and the challenges of maintaining alignment as AI systems evolve.

Chapter 5. Diffusion Models

This chapter will introduce diffusion models, a class of generative models that have shown remarkable proficiency in synthesizing high-quality data, especially text-to-image data. We begin by laying the theoretical groundwork, explaining the stochastic diffusion process that gradually adds noise to data and the reverse process that generates new data from noise. We will then discuss the architecture of UNet models, the backbone of many diffusion processes, detailing how their design enables the capture of multi-scale representations crucial for generating photorealistic images from textual descriptions. The lecture will also cover the latest innovations aimed at improving efficiency, such as the integration of autoencoder techniques for more resource-efficient training and generation.

Chapter 6. Computation and Memory Problems in Training Foundation Models

This lecture targets strategies for optimizing computation and memory usage in the training and finetuning of large foundational models. Topics include distributed training methodologies, memory-efficient techniques like ZeRO, Flash-Attention, and parameter-efficient finetuning, alongside explorations of mixture-of-experts architectures.

Chapter 7. Efficient Deployment Strategies for Foundation Models

This lecture will introduce popular techniques to reduce sizes of standard deep model and transformer models, such as quantization, pruning, knowledge distillation, their applications to model deployment, and the statistical rationales.

Chapter 8. Retrieval Augmented Generation

Examine how Retrieval-Augmented Generation (RAG) can improve the performance of LLMs in knowledge-intensive tasks, examining its computational scalability and safety.

Chapter 9. Safety in Generative AI

This lecture will introduce methods and metrics to assess the safety of generative models from two perspectives. One is from ethics perspective, including fairness and toxicity. We will also delve into content moderation techniques grounded in statistical detection and state-of-the-art watermarking techniques. The other is from machine learning security perspective. We will study several angles that practical AI systems must counteract, including adversarial examples, privacy, data poisoning backdoor attacks, membership inference attacks, model-stealing attacks, and their statistical foundations.

Chapter 10. Research Projects: Application Cases or New Methods

We will study the collected student-identified research problems, such as application cases or new methodological development.

In summary, this course aims to serve as a gateway to understanding the foundational ideas and exploring recent trends that drive the field. By offering hands-on experiences and highlighting theoretical bases, it will encourage students to apply their knowledge across scientific domains and foster innovative AI-for-science initiatives.

Homework:
Homework consists of weekly mandatory assignments as part of the course. All submissions must be typed and submitted either as Jupyter Notebooks or as executable Python files. Please ensure to include any dependency requirements. Additionally, clearly indicate the start or 'entry point' of your code, along with detailed instructions on how to run the files and the expected inputs and outputs. This will help ensure that your homework can be directly executed and evaluated without any setup issues.

Discussing homework with classmates is allowed, but submissions must be your own work. Conscientious completion of all homework assignments is essential to getting a good grade in this course (see "Grading" and "Academic Honesty" below).
No late homework will be accepted unless **prior** permission has been obtained from the instructor.

Grading:
Homework 40% (evenly distributed among all assignments); Final Project 40% (10% proposal, 10% presentation, 10% documented codes, and 10% written report); Participation and responsiveness 10%; Active team collaboration 10%; 10% Contribution to the course material 10% (Bonus). In general, 90% (out of the 100% score) guarantees "A" or "A-", 80% guarantees "B" or "B-", 65% guarantees "C", although the exact percentages will vary depending on the difficulty of the homework and exams. The same grading procedures or course requirements apply to graduate and undergraduate students. More grading policies can be found at https://onestop.umn.edu/academics/grading-policies .

Final Project Guidelines:
For the final project, students are required to collaborate in pairs. Teams should be self-organized and finalized by the end of Week 2. Once teams are formed, each group is expected to develop and submit a project proposal by the end of Week 6. Teams are encouraged to connect with a member of the course design team for guidance throughout their project development (including proposal), provided that the design team member volunteers to assist. Projects can be based on a list of suggested problems provided by the instructor or on any other topic of the students' choosing. If any student faces difficulties in forming a team, the course instructor will be available to help.

In the final two weeks of the course, each team will be allotted a 30-minute slot to present their project. These presentations will offer an opportunity to share findings and

methodologies with peers and to engage in constructive dialogue. Additionally, a comprehensive written report and associated open-source codes must be submitted. The report should encapsulate the team's research, results, and analytical discourse, serving as a formal documentation of their work.

Examples of a typical homework or default project include training or fine-tuning a small or medium-sized language model (e.g., GPT2, Llama-2-7B, Bloom) using novel domain-specific data, developing novel AI application cases, establishing benchmarks to evaluate large language models to understand their capabilities, limitations and risks, releasing new LLMs (1B+ parameters) with an interesting use scenario, creating new model architectures, and proposing new efficient training approaches.

Class Participation:
Active participation is encouraged. Essential information, including deadlines and exam details, will be shared in class or through video content.

Late Submission Policy:
Late submissions **of homework or final project** will be given ***only for documented reasons outside your control***, e.g., **emergency** conditions supported by a letter from a doctor. Noncritical health conditions and social/vacation conflicts are not acceptable reasons. There will be no makeup for the final project.
Incompletes: A grade of "I" will be given only in extraordinary circumstances, and then only by written agreement between the instructor and student. Students wishing to make up a prior incomplete must obtain permission from the instructor before the course starts.

Academic Honesty:
The following definition of student academic integrity and scholastic dishonesty is slightly modified from the webpage http://www.oscai.umn.edu:
Scholastic dishonesty means plagiarizing; cheating on assignments or examinations; engaging in unauthorized collaboration on academic work; taking, acquiring, or using test materials without faculty permission; submitting false or incomplete records of academic achievement; acting alone or in cooperation with another to falsify records or to obtain dishonestly grades, honors, awards, or professional endorsement; altering, forging, or misusing a University academic record; or fabricating or falsifying data, research procedures, or data analysis.

All School of Statistics teaching faculty are instructed to refer students who violate the policy for academic honesty and dishonesty to the Office for Community Standards.
A student responsible for scholastic dishonesty may also be assigned a penalty up to and including an "F" or "N" final grade for the course. If you have any questions regarding the expectations for a specific assignment or exam, ask.

The purpose of homework is to help you learn the material covered in this class. Many students find working on homework in groups to be helpful to their learning, and so this is specifically allowed. Even if you study and do homework in groups, however, your submitted solutions must include your own computer coding, computer output that you produced, and descriptions/summaries of results in your own words. Students who turn in identical papers or nearly identical papers are potentially guilty of academic dishonesty and may face sanctions. **Any submitted homework that contains a result from discussions must be annotated and acknowledged**. For example, this could be a sentence like "The xyz part of the submitted homework is based on a joint discussion with Alice and Bob" at the front line of submission.

University of Minnesota Accommodation Statement:

The University of Minnesota is committed to providing equitable access to learning opportunities for all students. Disability Resource Center (DRC) https://diversity.umn.edu/disability/ is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations. If you have, or think you may have, a disability (e.g., mental health, attentional, learning, chronic health, sensory, or physical), please contact DRC to arrange a confidential discussion regarding equitable access and reasonable accommodations.

University Policies:

More course related university policies can be found at http://www.policy.umn.edu/Policies/Education/index.htm#ctgeducation .