CSCI 8980 Topics in Modern Machine Learning

Fall 2021

General Information

This special topics course examines deep learning through the lens of classical and modern approximation, optimization, and learning theory, and targets PhD students with research focus/interest in the foundations of modern machine learning, data science, and artificial intelligence. Together with the instructor, the students are expected to 1) understand the existing theories for deep learning, and identify their strengths and limitations; 2) formulate and potentially solve new theoretical and methodological questions that are relevant to practice.

- **Prerequisite:** Course (CSCI5525 or equivalent, CSCI5980—Deep Learning) and research experience in machine learning and deep learning. Maturity in mathematics (linear algebra, multivariate calculus, probability, and numerical optimization), machine/deep learning is assumed. Ability to implement and test deep learning ideas in Tensorflow or Pytorch framework.
- **When & Where:** Mon/Wed 2:30–3:45pm, Amundson Hall 156 (synchronous on Zoom also)
- **Who:** Professor Ju Sun (Instructor) Email: jusun@umn.edu
- **Office Hours:** Mon/Wed 4–5pm (synchronous on Zoom also)
- **Course Website:** https://sunju.org/teach/TMML-Fall-2021/ All course materials, including course schedule, lecture slides and notes, project description, will be posted on the course website. Enrolled students are encouraged to check the website on a regular basis. Canvas is mostly used for announcement, discussion, and homework submission.
- **Communication—minimizing emails: Canvas** is the preferred and most efficient way of communication. Please post all non-confidential questions and discussions related to the course in Canvas instead of sending emails. If you have to use emails, please begin the subject line with “CSCI 8980", and expect delay in response—please send in reminders if facing significant delays.

Tentative Topics

Grouped by 6 key words as follow; selected reading for each topic will be posted on the course website. The list is subject to change.

- Approximation
- Optimization
- Generalization
- Invariance
- Robustness
- Generation

Teaching Format

Hybrid mode: Physical+Zoom synchronous attendance.

This is **not** to encourage remote attendance but to provide maximal possible flexibility in view of the evolving pandemic situation. **Caveat emptor**: We will use chalkboard presentation most of the time. We are not supported by a TA or UNITE; we connect to the builtin Audio/Video bar (`https://classroom.umn.edu/audiovideo-bar`) in the classroom to broadcast the class to Zoom. So the Zoom experience may be suboptimal compared to the classroom experience. We will also Zoom-record the lectures and release the videos on Media Gallery in Canvas. But attending the class, either in-person or remote, is highly recommended, as we expect active class discussion, which can be related to your weekly reports.

Mask policy for physical attendance: Proper masking is **required** for everyone in the classroom, **regardless of vaccination status**. Please check out this statement provided by the Faculty Consultative Committee (FCC) of the Faculty Senate and Senate Committee on Educational Policy for details, and the university COVID-19 guideline for updates.

Recommended References

There is no required textbook. Lectures and class discussions will mostly be based on recent papers that will be posted on the course website. References below are recommended resources covering part of the machine learning and mathematics foundations.

- Advanced machine learning textbooks
  - **Foundations of Machine Learning** (2e) by Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. `https://cs.nyu.edu/~mohri/mlbook/`
  - **Understanding Machine Learning: From Theory to Algorithms** by Shai Shalev-Shwartz and Shai Ben-David. `https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/`
  - **The Elements of Statistical Learning: Data Mining, Inference, and Prediction** by Trevor Hastie, Robert Tibshirani, and Jerome Friedman. `https://web.stanford.edu/~hastie/ElemStatLearn/`
  - **Probabilistic Machine Learning: An Introduction** by Kevin P. Murphy. `https://probml.github.io/pml-book/book1.html`
  - **Probabilistic Machine Learning: Advanced Topics** by Kevin P. Murphy. `https://probml.github.io/pml-book/book2.html`
  - **Patterns, Predictions, and Actions** by Moritz Hardt and Benjamin Recht. `https://mlstory.org/`
- Mathematics foundations
  - **Mathematics for Machine Learning** by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. `https://mml-book.github.io/`
  - **Linear Algebra and Optimization for Machine Learning** by Charu C. Aggarwal. `https://rd.springer.com/book/10.1007/978-3-030-40344-7` (UMN library access; login required)

- Deep learning foundations
  - **Think deep learning (course materials by Prof. Ju Sun)** https://sunju.org/teach/DL-Fall-2020/
  - **Deep Learning** by Ian Goodfellow and Yoshua Bengio and Aaron Courville. https://www.deeplearningbook.org/
- Hand-on in Python
  - **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems** by Aurélien Géron (2ed). O'Reilly Media, 2019. click here (UMN library access; login required)

Assessment

weekly reports $6\% \times 8$ + class participation 12% + final project: 40%

More on Week Reports

The weekly reports (≥ 500 words each; please include word count when submitting; math-heavy post can be submitted as PDF attachment) will be submitted and graded as Canvas discussion posts, and hence are visible to everybody in the class. Each report should reflect on the topics we cover in the week, and focus on one or several aspects of this list and beyond:

- discussing strengths and limitations
- any ways to test or even disprove the theory or methods
- results associated with above if you have produced any
- any new problems inspired and any results if you have produced any

In other words, **you are expected to be critical and creative**! You are strongly encouraged to comment on each others' posts and suggest improvements, and likewise revise your posts based on others' feedback; this will also improve your score. We strive together to penetrate the existing results, formulate new research problems, and even solve them.

Collaboration on the reports is strong encouraged! But you need to post your own reports. Since all submissions will be public,plagiarism and cheating can be easily identified, and are subject to disciplinary action. Please consult the student code of conduct for more information: https://regents.umn.edu/sites/regents.umn.edu/files/2019-09/policy_student_conduct_code.pdf

Course Project

The course project is to be performed by teams of 2 students. All students from the same team will get the same score for their course project.

Programming and Computing

Our programming environment will be Python 3. For deep learning frameworks, PyTorch (≥ 1.0) is preferred, but Tensorflow (≥ 2.0) is also accepted and supported. For small-scale experiments, Google Colab (https://colab.research.google.com/) and UMN MSI notebook service will suffice. Local installation of the relevant software packages may be a reasonable alternative. For large-scale course projects, we can use the Minnesota Supercomputing Institute (MSI) GPU computing queues based on our class account.

Other Policies

Please consult this policy link https://policy.umn.edu/education/syllabusrequirements-appa