CSCI 5980 (Sec 02) Think Deep Learning

Spring 2020

General Information

Over the last few years, deep neural networks (DNN) have fundamentally transformed the way people think of machine learning and approach practical problems. Successes around DNN have ranged from traditional AI fields such as computer vision, natural language processing, interactive games, to health care, physical sciences—touching each and every corner of theoretical and applied domains. On the other hand, DNN still largely operate as black-boxes and we only have very limited understanding as for when and why they work. This course introduces basic ingredients of DNN, samples important applications, and throws around open problems. Emphasis is put on thinking from first principles, as the field is still evolving rapidly and there is nothing there that cannot be changed.

- **Prerequisite:** Introduction to machine learning or equivalent. Maturity in linear algebra, calculus, and basic probability is assumed. Familiarity with Python (esp. numpy, scipy) is necessary to complete the homework assignments and final projects.
- **When & Where:** T/Th 2:30PM – 3:45PM, Akerman Hall 225
- **Who:**

  Professor Ju Sun (Instructor) Email: jusun@umn.edu
  Yuan Yao (TA) Email: yaoxx340@umn.edu
  Taihui Li (Courtesy TA) Email: lixx5027@umn.edu

- **Office Hours:** Professor Ju Sun (Instructor) Th 4 – 6pm at 5-225E Keller H
  Yuan Yao (TA) Wed 12:15 – 2:15pm at Shepherd Lab 234
- **Course Website:** https://sunju.org/teach/DL-Spring-2020/ All course materials, including course schedule, lecture notes, homework assignments, project description, and additional notes will be posted on the course website. Enrolled students are encouraged to check the website on a regular basis.
- **Communication: Canvas** is the preferred and most efficient way of communication. Please post all 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 5980".

Tentative Topics

There will be 21 lecture sessions, each running 75 minutes. The tentative topics and time allocation are as follows.

- Course overview (1)
- Neural networks: old and new (1)
- Fundamental belief: universal approximation theorem (2)

- Numerical optimization with math: optimization with gradient descent and beyond (2)
- Numerical optimization without math: auto-differentiation and differential programming (2)
- Work with images: convolutional neural networks (2)
- Work with images: recognition, detection, segmentation (2)
- To train or not? scattering transforms (2)
- Work with sequences: recurrent neural networks (2)
- Learning probability distributions: generative adversarial networks (2)
- Learning representation without labels: dictionary learning and autoencoders (1)
- Gaming time: deep reinforcement learning (2)

The lecture sessions are interlaced with 5 discussion sessions, each also 75 minutes, that are designed to help the students master the critical computational and practical components and successfully complete their homework assignments and final projects.

- Python, Numpy, and Google Cloud/Colab
- Project ideas
- Tensorflow 2.0 and Pytorch
- Backpropagation and computational tricks
- Research ideas

Recommended References

There is no required textbook. Lecture notes and additional notes will be the primary resources. Recommended reference books are

- **Deep Learning** by Ian Goodfellow and Yoshua Bengio and Aaron Courville. MIT Press, 2016. Online URL: `https://www.deeplearningbook.org/` (comprehensive coverage of recent developments)

- **Neural Networks and Deep Learning** by Charu Aggarwal. Springer, 2018. UMN library online access (login required): Click here. (comprehensive coverage of recent developments)

- **The Deep Learning Revolution** by Terrence J. Sejnowski. MIT Press, 2018. UMN library online access (login required): Click here. (account of historic developments and related fields)

- **Deep Learning with Python** by François Chollet. Online URL: `https://livebook.manning.com/book/deep-learning-with-python` (hands-on deep learning using Keras with the Tensorflow backend)

- **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. UMN library online access (available soon). (hands-on machine learning, including deep learning, using Scikit-Learn and Keras)

Assessment

- Homework 60%: 7 Homeworks, the top 5 scored will count
- Course project 40%: proposal (5%) + mid-term presentation (10%) + final report (25%).
- Exception: If the course project leads to a publishable result as determined by the instructor, the participating students will all automatically get an A for the course.

Homework

You have approximately two-week time to complete each homework (exact due date will be specified in each assignment). No late submissions will be accepted. All submissions *must* be electronic and uploaded via the Canvas system. Written part *must* be typeset in LaTeX and submitted as PDF files. Computer programs *must* be submitted in the Python notebook format. Only Python 3 will be used and accepted in this course.

Collaboration on homework problems is strongly encouraged, but each student must ensure that the final submission is prepared individually. *Collaborators should be properly acknowledged in the final submission, at the problem level.* The same applies to computer programs. Plagiarism and cheating will not be tolerated 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 or 3 students and the weight of the project should be proportional to the number of students in the team. All students from the same team will get the same score for their course project.

The project can be but not limited to survey of literature on a focused topic, implementation and comparison of existing methods, novel application of DNN techniques, and development of new methods and theories. The instructor will throw around project ideas throughout the course, and will also provide detailed feedback to the project proposal and mid-term presentation.

Programming and Computing

Our programming environment will be Python 3. For deep learning, both Tensorflow (≥ 2.0) and PyTorch (≥ 1.0) will be accepted and supported. For small-scale experiments, which will be the case for homework assignments, Google Colab (https://colab.research.google.com/) and Google Cloud (each enrolled student gets $50 free credits) will suffice[1]. Also, local installation of the relevant software packages may be a reasonable alternative. For course projects that needs to scale up, we can arrange resources from Minnesota Supercomputing Institute (MSI) which has recently significantly expanded their GPU computing queue.

[1] We also get similar free credits from AWS.

Related Courses

Within UMN

- **Topics in Computational Vision: Deep networks** (Prof. Daniel Kersten, Department of Psychology. Focused on connection with computational neuroscience and vision)
- **Analytical Foundations of Deep Learning** (Prof. Jarvis Haupt, Department of Electrical and Computer Engineering. Focused on mathematical foundations and theories)

Global

- **CS230 Deep Learning** (`https://cs230.stanford.edu/`, Stanford Computer Science)
- **CS231n: Convolutional Neural Networks for Visual Recognition** (`http://cs231n.stanford.edu/`, Stanford Computer Science, 2019)
- **CS224n: Natural Language Processing with Deep Learning** (`http://web.stanford.edu/class/cs224n/`, Stanford Computer Science, 2020)
- **Analyses of Deep Learning** (`https://stats385.github.io/`, Stanford Statistics, 2019)
- **Advanced deep learning and reinforcement learning** (`https://github.com/enggen/DeepMind-Advanced-Deep-Learning-and-Reinforcement-Learning`, UCL/Deepmind, 2018)
- **Mathematics of Deep Learning** (`https://joanbruna.github.io/MathsDL-spring18/`, NYU Courant Institute, 2018)
- **MIT Deep Learning**(`https://deeplearning.mit.edu/`, MIT courses and lectures on deep learning, deep reinforcement learning, autonomous vehicles, and artificial intelligence)
- **CMSC 35246 Deep Learning** (`https://ttic.uchicago.edu/~shubhendu/Pages/CMSC35246.html`, U Chicago Computer Science, 2017)
- **Neural Networks for Machine Learning** by Jeof. Hinton (`https://www.cs.toronto.edu/~hinton/coursera_lectures.html`, U Toronto, 2012)