CSCI 1470, 2470: Deep Learning

Course Syllabus

Fall 2021

Note: CSCI 2470 shares the same course staff, lectures, labs, and assignments as 1470, but with the addition of weekly homework sets and a more advanced final project. 1470 is open to everyone who meets the prerequisites. 2470 is a graduate-level course and is intended for graduate students; registration is gated accordingly. The instructor reserves the right to admit undergraduates into 2470 at his discretion.

Lecture Time and Location:

MWF 12:00-12:50 PM Eastern Time
Classroom TBD, website sites.google.com/brown.edu/cs1470-2021-fall
Lectures will be given live and recorded for asynchronous viewing.

Course Staff:

Professor	Chen Sun (csun45)
HTA	Colby Anderson (cander23)
	Usha Bhalla (ubhalla)
	Linghai Liu (lliu58)
UTA	Chace Hayhurst (chayhurs)
	Damian Wasilewicz (dwasilew)
	David Lee (ilee26)
	Jack Dermer (jdermer)
	Jacob Yu (zyu28)
	Jason Ma (jma14)
	Jason Manuel (jmanuel2)
	Jack Murphy (jmurph17)

	Justin Sanders (jsander8)
	Luca Fonstad (lfonstad)
	Mary Lou (mlou2)
	Mason Zhang (mzhan104)
	Neal Mahajan (nmahaja1)
	Sally Zhi (szhi)
	Tian Yun (tyun)
UTA, STA	Amanda Lee (zlee8)
	Erika Bussmann (ebussman)

Introduction:

Welcome to CSCI 1470 / 2470! Deep Learning belongs to a broader family of machine learning methods. It is a particular version of artificial neural networks -- a version that emphasizes learning representation with multiple layers of networks. Deep Learning, plus the specialized techniques that it has inspired (e.g. convolutional neural networks, recurrent neural networks, and transformers), have led to rapid improvements in many applications, such as computer vision, machine learning, sound understanding, and robotics. This course intends to give students an overview of the prominent techniques of Deep Learning and its applications in computer vision, language understanding, and other areas. It also aims at providing hands-on practice of implementing deep learning algorithms in Python.

Learning Goals:

Students who complete this course will:

- Learn the basic concepts and tools that underlie all modern deep learning systems (e.g. back-propagation and automatic differentiation).
- Know which model architectures to use for processing different types of data (images, sequences, and graphs).
- Grow hands-on experience implementing models for vision, language, or robotics applications.

- Be practiced in critically analyzing the potential societal impacts of the deep networks they develop.
- Team up and implement an existing research paper or something new.

Lectures:

Sept. 8, Wed.	*Rosh Hashanah, no class*
Sept. 10, Fri.	Introduction to deep learning
Sept. 13, Mon.	Perceptrons
Sept. 15, Wed.	Loss functions and optimization
Sept. 17, Fri.	Backpropagation
Sept. 20, Mon.	Backpropagation, automatic differentiation
Sept. 22, Wed.	Introduction to convolutional neural networks
Sept. 24, Fri.	Deep learning hardware and softwares
Sept. 27, Mon.	Introduction to Tensorflow
Sept. 29, Wed.	The life cycle of machine learning systems
Oct. 1, Fri.	Convolutional neural networks, from MNIST to ImageNet (1)
Oct. 4, Mon.	Convolutional neural networks, from MNIST to ImageNet (2)
Oct. 6, Wed.	Convolutional neural networks in the real world
Oct. 8, Fri.	Language models, word embeddings
Oct. 11, Mon.	*Indigenous Peoples' Day, no class*
Oct. 13, Wed.	Feedforward language models
Oct. 15, Fri.	Recurrent neural networks
Oct. 18, Mon.	LSTMs
Oct. 20, Wed.	Machine translation
Oct. 22, Fri.	Transformers (1)
Oct. 25, Mon.	Transformers (2)

Oct. 27, Wed.	Self-supervised learning
Oct. 29, Fri.	Scaling deep learning systems
Nov. 1, Mon.	Variational Autoencoders
Nov. 3, Wed.	Generative adversarial networks
Nov. 5, Fri.	Adversarial attacks, Deep fakes
Nov. 8, Mon.	Responsible deep learning
Nov. 10, Wed.	Graph neural networks (1)
Nov. 12, Fri.	Graph neural networks (2)
Nov. 15, Mon.	Introduction to reinforcement learning (1)
Nov. 17, Wed.	Introduction to reinforcement learning (2)
Nov. 19, Fri.	Deep Q learning
Nov. 22, Mon.	Policy gradient methods
Nov. 24, Wed.	*Thanksgiving, no class*
Nov. 26, Fri.	*Thanksgiving, no class*
Nov. 29, Mon.	Actor-critic methods
Dec. 1, Wed.	Case study: self-driving
Dec. 3, Fri.	Guest lecture: self-supervised learning (TBD)
Dec. 6, Mon.	Guest lecture: multimodal learning (TBD)
Dec. 8, Wed.	Guest lecture: 3D deep learning (TBD)
Dec. 10, Fri.	Conclusion
Dec. 13, Mon.	Final project presentations

The lectures are not yet finalized and subject to change.

Prerequisites:

- A basic programming course: (CSCI 0150, 0170 or 0190)

- A linear algebra course: (CSCI 0530, MATH 0520 or 0540)
- A stats / probability course: (CSCI 0220, 1450, 0450, MATH 1610, APMA 1650 or 1655)

Exceptions may be possible for those missing one of these prerequisites if (a) the student has taken another course which covers similar material, or if (b) the student will be concurrently taking the prerequisite. If either of these situations applies to you, use the "Request Override" feature in Courses@Brown to request an override (and explain why you believe your situation merits one).

Grading:

There are seven assignments, and a final project. The grade breakdown is:

- 70% Assignments
- 30% Final project

Textbook:

None required. Students are encouraged to refer to the following textbooks, all of which are available online:

- Deep Learning, by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
- Probabilistic Machine Learning: An Introduction, by Kevin Murphy.
- Patterns, Predictions, and Actions: A Story about Machine Learning, by Moritz Hardt, and Benjamin Recht.

Assignments:

Assignments consist of a programming component (e.g. implementing and training some neural network model) as well as a written component. The written component contains both algorithmic conceptual questions as well as an open-ended ethics question that requires students to reflect on societal ramifications of the machine learning models they are building in the programming portion of the assignment. Answers to the written questions will be due before the programming component is due, i.e. there are two separate deadlines: one for the written component, followed by one for the programming component.

Late Policy: Each of the assignments is due by 11:59pm Eastern Time on the given due date — thus, you have until the end of the day listed as the due date to complete the assignment. You will have four (4) free late days to use towards all but the last project.

After your late days are expended you will lose 10% of your project grade for each extra day your handin is delayed. At the end of the term, we will calculate how to best divvy out your late days to best help your final grade. There is no need for you to do anything special for this, other than keep track of submission dates for your own records. Note that because we do this, use of late days will not be reflected in the initial grade report for your assignments.

Exceptions to the Late Policy: Sometimes there are special circumstances during the semester that result in exceptions to this late policy. All such circumstances require an official note from the Deans. In general, they only provide support notes on behalf of students who are experiencing disruptive medical or personal circumstances, including those related to Title IX situations, that affect their ability to do academic work in a timely way. If you believe an extension is warranted for you, and you have a note from the Deans, please notify the instructor by submitting an Extension Request Form (which can be found on the "Resources" page of the course website). What you should not do:

- Do not send email to the instructor requesting an extension—you will not receive a response.
- Do not send a dean's note to a UTA/HTA. These often contain sensitive personal information, the receipt of which by anyone other than the instructor is a FERPA violation.

You should manage other special circumstances such as interviews, personal travel or extra-curricular factors using the late day policy above.

Final Project:

All students will be required to complete a final project in groups of 2-3. At minimum, this should entail re-implementing the methods described by a recent deep learning research paper. The course staff will provide resources to help students find and select appropriate papers for their projects.

Expectations are **higher** for 2470 students and 1470 capstone students: they will be required to go above-and-beyond re-implementing an existing research paper, and will be required to attempt a novel research project. This could involve investigating a new model architecture to solve an existing problem, adapting methodology from a different problem to solve an existing problem, or defining and solving a new problem of interest.

Programming:

All programming in this course will be done in Python, primarily using numpy and the Python Tensorflow API. Labs will be provided as Google Collaboratory notebooks. Since this is not a software engineering course, we won't be enforcing stringent style guidelines, but you should write so that someone who isn't a Python wizard will be able to understand what your program is doing (add plenty of comments, break up code into smaller functions, i.e. apply basic common sense). If you turn in a partially-functional assignment and we can't tell what you were trying to do, we'll probably be very grumpy about giving partial credit.

As that translates to an official policy, so long as your code produces the expected output(s) and adheres to any specific project restrictions (runtime, etc.) then you will not lose points for poor design or coding practices. However, as this is not a software design course, it is not the responsibility of the TAs to attempt to understand the intentions underlying confusing code. If it is not fully clear what you were trying to do in the implementation of a partially-functional assignment (i.e. not all of the output is as expected) then partial-credit will be given sparingly and at our discretion.

Capstone:

This course may be used as a capstone course for an Sc.B. degree. Interested students should email a filled-out copy of the capstone form to the instructor for him to sign and return. All 1470 students who use the course as a capstone will be expected to fulfill the same final project requirements as students taking 2470. Accordingly, 1470 capstone students may form final project groups with 2470 students.

Academic Integrity & Collaboration Policy:

Academic dishonesty will not be tolerated. This includes cheating, lying about course matters, plagiarism, or helping others commit a violation. Plagiarism includes reproducing the words of others without both the use of quotation marks and citation. Students are reminded of the obligations and expectations associated with the Brown Academic and Student Conduct Codes.
Discussion of course material with your classmates is both permitted and encouraged. However, showing, copying, or other sharing of actual code or verbatim answers to written questions is forbidden. This includes publishing projects on Github or any other public platform. This policy **will** be enforced.

One of the ways that we enforce the collaboration policy is by running MOSS on all code submissions. For those who are new to the department or otherwise unfamiliar with it, MOSS (short for "measure of software similarity") is a software tool which detects similarities between pieces of code. The course staff manually examines the output of MOSS to look for cases where two students' submitted code is similar in such a way that it is sufficiently unlikely for them to have independently produced it (i.e. an instance of cheating or code plagiarism).

Diversity & Inclusion:

Our intent is that this course provides a welcoming environment for all students who satisfy the prerequisites. Our TAs have undergone training in diversity and inclusion, and all members of the CS community, including faculty and staff, are expected to treat one another in a professional manner. If you feel you have not been treated in a professional manner by any of the course staff, please contact either the instructor, Ugur Cetintemel (Dept. Chair), Tom Doeppner (Vice Chair) or Laura Dobler (diversity & inclusion staff member). We will take all complaints about unprofessional behavior seriously.

Brown welcomes students from all around the country and the world, and their unique perspectives enrich our learning community. To empower students whose first language is not English, an array of support is available on campus, including language and culture workshops and individual appointments. For more information, contact the English Language Learning Specialists at ellwriting@brown.edu.

Accomodations:

Brown University is committed to full inclusion of all students. Please inform the instructor if you have a disability or other condition that might require accommodations or modification of any of these course procedures. You may email the instructor, come to office hours, or speak with him after class, and your confidentiality is respected. We will do whatever we can to support accommodations recommended by SEAS. For more information contact Student and Employee Accessibility Services (SEAS) at 401-863-9588 or SEAS@brown.edu. Students in need of short-term academic advice or support can contact one of the deans in the Dean of the College office.

Mental Health:

Being a student can be very stressful. If you feel you are under too much pressure or there are psychological issues that are keeping you from performing well at Brown, we encourage you to contact Brown's Counseling and Psychological Services (CAPS). They provide confidential counseling and can provide notes supporting extensions on assignments for health reasons.

Incomplete Policy:

We expect everyone to complete the course on time. However, we certainly understand that there may be factors beyond your control, such as health problems and family crises, that prevent you from finishing the course on time. If you feel you cannot complete the course on time, please discuss with the instructor the possibility of being given a grade of Incomplete for the course and setting a schedule for completing the course in the upcoming year.

Acknowledgments:

Thanks to Tom Doeppner and Laura Dobler for the text on accommodation, mental health, and incomplete policy. This document was adapted from an earlier version written by Daniel Ritchie.