Course Missive

Fall 2020

Lecture Time and Location:
MWF 12:00-12:50 PM US Eastern Time
Lecture given live and recorded for asynchronous viewing.
Live participation welcome but not required.

Course Staff

What	Who	Where	When
Professor	Daniel Ritchie (`dritchi1`)	TBA	TBA
HTAs	Bryce Blinn (`bblinn`)	TBA	TBA
	Kevin Du (`kdu3`)	TBA	TBA
	George Lee (`glee43`)	TBA	TBA
	Koyena Pal (`kpal1`)	TBA	TBA
UTAs.	Andrew Canino (`acanino`)	TBA	TBA
	Amy Pu (`apu1`)	TBA	TBA
	Antony Sagayaraj (`asagayar`)	TBA	TBA
	Arvind Yalavarti (`ayalava2`)	TBA	TBA
	Ben Silverman (`bsilverm`)	TBA	TBA
	Cindy Li (`cli94`)	TBA	TBA
	Dan Smits (`dsmits`)	TBA	TBA
	Eliot Laidlaw (`elaidlaw`)	TBA	TBA
	Emily Reed (`ereed2`)	TBA	TBA
	Greyson Gerhard-Young (`ggerhard`)	TBA	TBA
	Griffin Kupsaw (`gkupsaw`)	TBA	TBA
	Gaby Rizk (`grizk`)	TBA	TBA
	Harman Suri (`hsuri`)	TBA	TBA
	Isaac Hilton-VanOsdall (`ihiltonv`)	TBA	TBA
	Jason Manuel (`jmanuel2`)	TBA	TBA
	Kei Nawa (`knawa`)	TBA	TBA
	Kyle Qian (`kqian2`)	TBA	TBA
	Katherine Sang (`ksang1`)	TBA	TBA
	Kotone Ninagawa (`ktsuji`)	TBA	TBA
	Gene Siriviboon (`psirivib`)	TBA	TBA
	Qingyi Lu (`qlu9`)	TBA	TBA
	Raymond Cao (`rcao6`)	TBA	TBA
	Steven Cheung (`scheung9`)	TBA	TBA
	Sonny Mo (`smo1`)	TBA	TBA
	Suhye Park (`spark108`)	TBA	TBA
	Sally Zhi (`szhi`)	TBA	TBA
	Tucker Berckmann (`tberckma`)	TBA	TBA
	Tyler Jiang (`tjiang12`)	TBA	TBA
	Top Piriyakulkij (`wpiriyak`)	TBA	TBA
	Willem Speckmann (`wspeckma`)	TBA	TBA
	Wenhuang Zeng (`wzeng3`)	TBA	TBA
	Joy Zheng (`zzheng6`)	TBA	TBA
	Qian Zhang (`qzhang93`)	TBA	TBA
ETAs	Dybe Mwaisyange (`dmwaisya`)	TBA	TBA
	Naomi Lee (`nlee16`)	TBA	TBA

Introduction

Welcome to CS 1470/2470! Over the past few years, Deep Learning has become a popular area, with deep neural network methods obtaining state-of-the-art results on applications in computer vision (Self-Driving Cars), natural language processing (Google Translate), and reinforcement learning (AlphaGo). These technologies are having transformative effects on our society, including some undesirable ones (e.g. deep fakes). This course intends to give students a practical understanding of how Deep Learning works, how to implement deep neural networks, and how to apply them ethically. We introduce students to the core concepts of deep neural networks, including the backpropagation algorithm for training neural networks, as well as specific operations such as convolution (in the context of computer vision) and word embeddings and recurrent neural networks (in the context of natural language processing). Throughout the lectures, labs, and assignments, we emphasize and require students to think critically about potential ethical pitfalls that can result from misapplication of these powerful models. The course is taught using the Tensorflow deep learning framework.

Learning Goals

Students who complete this course will:

- Understand the fundamental algorithms that underly all modern deep learning systems (backpropagation and automatic differentiation)
- Be able to implement deep learning models in Tensorflow
- Know which model architectures to use for processing different types of data (images, sequences, and graphs)
- Have familiarity with training neural networks through supervised learning, unsupervised learning, and reinforcement learning
- Be practiced in critically analyzing the potential societal impacts of the deep networks they develop

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 code (and explain why you believe your situation merits one).

Coursework

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

- 65% – Assignments (programming + written components)
- 15% – Weekly Labs
- 20% – Final Project

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 AoE (Anywhere on Earth) Time[1] 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. Please send all such notes to the instructor. ***Do not send these notes to the HTAs!*** Extension requests 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.

Labs

Students will be assigned to an hour-long weekly lab slot; these slots will be determined based on a poll of student availability to be conducted early in the semester. It is expected that students

[1] `https://time.is/Anywhere_on_Earth`

should be able to complete the exercises for a given lab, and get them checked off by the lab TA, during this assigned slot. Lab materials for each week will be released online at the beginning of the week; students are welcome to look at and beginning working on the lab in advance of their assigned lab slot, if they wish. If the student does not complete the lab and/or get it checked off during their assigned slot, they can go to another lab section that week to get it checked off (without penalty).

If a student missess their assigned lab slot, they can also go to another lab section to get the lab checked off. However, this will incur a lab grade penalty (e.g. a 3/4 instead of a 4/4 for that lab). Students will be permitted to miss their assigned lab slot three (3) times without incurring this penalty.

In all cases, a lab must be checked off at latest by the student's assigned lab slot the following week.

If a student knows well in advance that they will not be able to make their assigned lab slot but will be able to make another slot that week, the student may message their lab TA ***at least 48 hours in advance*** to be temporarily switched into a different lab slot for that week. This will not incur a late lab penalty.

Lab collaboration Students may complete labs in groups of two, if they wish. In this case, both students must be present for the TA to check them off, and the TA will ask check-off questions of both students.

Final Project

All students will be required to complete a final project in groups of 2-4. At minimum, this should entail re-implementing the methods described by a recent deep learning research paper. Expectations are higher for 2470 students and 1470 capstone students; see the relevant sections below for more information.

The course staff will provide resources to help students find and select appropriate papers for their projects. In addition, we will invite "project pitches" from research groups and other organizations at Brown who are looking for deep learning help with some of their projects / data.

Deep Learning Day: The class's collective final project efforts culminate in "Deep Learning Day," an end-of-semester celebration consisting of poster sessions and oral presentations organized as a day-long symposium. **This year, Deep Learning Day will be organized as a virtual conference** (details TBD).

Differences Between 1470 and 2470

Both sections of this course share the same course staff, lectures, labs, and assignments. 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.

In terms of required coursework, the two sections have the following differences:

Assignments (Programming): Every programming assignment has a quantitative metric threshold that all students' code is expected to meet (e.g. test-set accuracy). The threshold for 2470 students is higher. Reaching this higher threshold may require more advanced model architectures, the implementation of which may require reading additional reference material.

Assignments (Written): 2470 students will be required to answer additional written questions. These may be mathematical (e.g. proving/deriving some property of a model/algorithm) or conceptual in nature (e.g. requiring reading a related paper and connecting its findings to what we have discussed in class).

Final Project: 2470 students will be required to go above-and-beyond re-implementing an existing research paper: rather, they 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.

Time Requirements

In addition to 3 hours per week in class, you will probably need 1 hour of help from a UTA, 1 hour for lab, 2 hours of reading and review to solidify your grasp of the material, and 6 hours for the assignments. (182 hours/semester)

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[2] 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[3].

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).

Regret clause: We recognize that college is stressful, and sometimes that stress leads to panicked, bad decision making near deadlines. We are also aware that this problem is likely to be exacerbated by the unusually stressful period of history through which we are all living right now. In the spirit of this understanding, we are implementing a "regret clause" for the course this year. If a student submits an assignment that they know to contain plagiarized material (either code or answers to written questions), that student may invoke the regret clause by informing the instructor of the plagiarized material *within 48 hours of the assignment submission deadline.* The instructor will ask to meet with the student, and the outcome of this meeting can vary. On a first invocation of the regret clause, the instructor will exercise his discretion, applying a penalty up to receiving zero credit on the plagiarized portion of the assignment. On subsequent invocations, all plagiarized portions of the assignment will automatically receive zero credit. In all cases, the occurrence of plagiarism will be kept confidential by the instructor.

If the course staff detects a plagiarized submission, and the student has not invoked the regret clause, then we will write up a formal plagiarism case to be heard by the Academic Code Committee. These hearings can result in a variety of outcomes, including immediately failing the course and receiving a "Directed NC" (i.e. a form of NC that *does* show up on your transcript, along with a note that the NC was given for an academic code violation).

[2] https://drive.google.com/file/d/1YYK7u4ccB0II52yxXiEI6sMwA7LZMrGy/view

[3] https://www.brown.edu/academics/college/orientation/academic-student-conduct-codes

Diversity & Inclusion

Our intent is that this course provide 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. Prof. Krishnamurthi has good notes on this area[4]. To access student support services and resources, and to learn more about diversity and inclusion in CS, please visit the Brown CS webpage[5].

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.

[4] `http://cs.brown.edu/courses/cs019/2016/professionalism.html`

[5] `http://cs.brown.edu/about/diversity/resources/`

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 Eugene Charniak.