CS 1470 Course Missive Spring 2026

Deep Learning

Instructor: Randall Balestriero

Time: M/W/F 12 pm-12:50 pm, Location: Salomon Center 001

Course Website: https://browndeeplearning.com/

Contact Information

Professor: randall_balestriero@brown.edu

HTAs: cs1470headtas@lists.brown.edu

All TAs: cs1470tas@lists.brown.edu

Introduction

Welcome to CSCI 1470! Over the past few years, Deep Learning has become a popular and influential 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), 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 the misapplication of these powerful models. The course is taught using the PyTorch deep learning framework.

Course Objectives

By the end of this course, you will be able to:

- Learn about the fundamental algorithms that underlie all modern deep learning models.
- Implement different types of deep learning models in PyTorch.
- Think critically about using a deep learning model for a task and its potential societal impact.
- Collaborate with classmates on a team project to apply deep learning models to task of your choice.

- Communicate your findings (both positive and negative results are encouraged) through presentations.

Prerequisites

- A basic programming course: (CSCI 0190 or 0200)
- A linear algebra course: (CSCI 0530, MATH 0520 or 0540)
- A stats/probability course: (CSCI 0220, 1450, 0450, MATH 1610, APMA 1650 or 1655)
- An AI/ML course: (CSCI 0410, 1411, 1420, 1950F, 1430 or 1460)

Exceptions may be possible for those missing one of these prerequisites if (a) the student has taken another course that 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).

Textbook

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

- Deep Learning, by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
- Dive into Deep Learning, by Aston Zhang, Zack C. Lipton, Mu Li, and Alex J. Smola.

Assessment of Learning

There are five larger homework assignments, 3 smaller mini-projects, and a final project. The grade breakdown for this course is as follows:

- 55% – Assignments (programming + written components)
- 20% – 3 workshop or SRC attendances (1 of each type)
- 20% – Final Project
- 5% – Class participation (assessed through weekly quizzes)

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 the societal ramifications of the machine learning models they are building in the programming portion of the assignment. Answers to the written questions are often due before the programming component deadline, i.e., there are two separate deadlines: one for the written component, followed by one for the programming component.

The assignments differ in difficulty and estimated time to complete and will be graded accordingly. We will use the following breakdown to weight assignment grades:

Assignment Number	Name	Weighting	Deadline
1	Setup and Mathematical Foundations	2.5	1/29
2	Intro to NP and TF	2.5	2/5
3	Beras	12.5	2/26
4	CNNs	10	3/12
5	Image Captioning	10	3/26
6	Generative Modeling	7.5	4/9
7	RL	10	4/23
Total	**Assignment Total**	**55**	

Late Policy

You will be given four free late days for use over the course of the semester for assignments. You can use these whenever you want and do not need to notify staff that you are using them. One late day will be used for any assignment submitted after the initial deadline, but before the 24 hours after the deadline. Two late days will be used if the assignment is submitted before 48 hours, and so on.

If you run out of late days, your grade for the late assignment will be reduced by 10% if submitted within 24 hours of the deadline, or 25% if submitted within 48 hours of the deadline. For most assignments, we will not accept submissions after 48 hours from the initial deadline unless an extension is granted by the professor. We will apply the free late days to the assignments that will benefit your final score the most.

SRC Discussion Sections

It is impossible to learn about deep learning and not discuss the societal impact of the models and their applications. With every passing year, SRC questions and issues related to deep learning are becoming more prevalent, and in this course, we would like to keep the dialogue going. Therefore, this semester, we will be holding ~6 SRC discussion sessions. Each student will sign up to attend 1 discussion session during the course. The UTAs will facilitate discussions related to the SRC topics covered in the homework assignments, and the students will participate in groups. More details will be provided during the course.

Workshops

There are certain topics and skills that you may find helpful while preparing to work on your final project of performing research with Deep Learning that we do not have time to cover in class. We will be offering workshops run by TAs throughout the semester that will be interactive ways to explore these topics. Some example workshops include: Introduction to Pytorch and Jax, how to read and implement a research paper, and deep learning applications, like deep learning for genomics. Much like the SRC discussions, you must attend at least one workshop out of your three sessions.

You are required to attend a TOTAL of at least 3 SRC or Workshop sessions during the semester. This total must include at least ONE SRC and at least ONE Workshop.

Interactive Grading

Every student is graded on at least 2 assignments total at random selection throughout the semester. However, you should be prepared for live 1-on-1 interactive grading sessions with a TA for any given project. During these sessions, you will be asked to explain your implementation details, reasoning, and specific function behaviors to demonstrate mastery of your code. You will be graded on a pass/fail basis. We are looking to see that students have understood the code that they have submitted. Students who fail the grading session will have their assignment flagged for additional review and/or receive a **20% penalty** on their overall project score.

Please note: this is designed to verify your understanding and confidence in your work, not a high-pressure interview or formal thesis defense.

Final Project

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

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. Details of the day will be shared during the course.

Students will be able to optionally opt into using LLMs to help with their final project. Groups that opt to use GenAI tools will have significantly higher expectations in terms of scope, production, and presentation.

Class Participation

Students are highly encouraged to attend the classes in person. However, we will record lectures for students to view if they miss a class or need to review class material. Since

keeping track of in-class participation is difficult in large classes, after Wednesday's lecture, the instructor will post a short quiz on Canvas. Students are expected to complete the short quiz by noon the next day (Thursday) in order to obtain full participation credit for the week. If you attend the class (or watch the lectures) regularly, you should be able to finish the quiz quickly and with little effort. The quiz will also help you review the course material regularly and make sure you are keeping up with the course topics. The students will be given full points for correct answers, partial points for incorrect answers, and no points for incomplete or no submission. Please note that no extensions will be provided for completing the quizzes. They contribute a relatively small percentage to your final grade, so missing a quiz or two will not affect your grade drastically.

How Can You Do Well?

This class has a high-level course load, and you can ensure your success in it by doing the following:

- Regularly attending classes and lab sections. Participation in class discussions is highly encouraged.
- Starting the work on assignments and projects early.
- Completing and turning in all assignments and quizzes on time.
- Equally contributing to the final project and clearly presenting your project.

Programming

All programming in this course will be done in Python, primarily using numpy and the Python PyTorch API. Labs will be provided as Google Colab 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.

Policy on the Use of AI-Powered Tools for Course Assignments

All work that students submit during the course must be their own original work and represent their own thoughts and ideas. As such, the use of AI-powered tools (such as OpenAI's ChatGPT or GitHub's CoPilot) for completing course assignments is discouraged. **The use of AI-powered tools without citation will be considered academic misconduct.** For programming assignments, generating code from a comment or "todo" statement provided in the stencil will also be considered academic misconduct. Pasting the assignment and code into an AI tool is academic misconduct. Since intelligent autocomplete tools, like GitHub Copilot, make it difficult to document usage, they are forbidden unless stated otherwise on a specific assignment.

If a student chooses to use these tools for course assignments, they must acknowledge and thoroughly document their use of the tool. The student must: 1) cite the tool used, 2) include an explanation of how the tool was used for the assignment, and 3) fully document the student's own contribution versus the contribution of the tool (e.g., including full ChatGPT transcripts as an appendix to your assignment). All assignments will be graded based on the student's original ideas – students risk losing credit if the documentation provided is insufficient to determine the student's original contributions.

There will be assignments (and portions of assignments) that will allow the use of AI tools. We will specify which portions of the assignments are acceptable to use AI. Final project groups will be able to opt into using AI tools with different expectations for their work.

Time Requirements

In addition to 3 hours per week in class, you will probably need 1 hour of help from a UTA, 2 hours of reading and review to solidify your grasp of the material (including submitting the weekly quiz), and 8 hours for the assignments. Additionally, you will need 2 hours (during the semester) for attending the 2 SRC discussion sessions and 2 hours for workshops. (184 hours/semester)

Capstone

This course may be used as a capstone course for an Sc.B. degree. Bring a copy of the capstone form to the instructor after class or during office hours.

Diversity Statement

This course is designed to support an inclusive learning environment where diverse perspectives are recognized, respected, and seen as a source of strength. It is our intent to provide materials and activities that are respectful of various levels of diversity: mathematical background, previous computing skills, gender, sexuality, disability, age, socioeconomic status, ethnicity, race, and culture.

Academic Integrity and 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 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 that 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).

Accommodations

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.

CSCI 1470 is now offered in the Fall and Spring semesters with similar course content. Therefore, starting this academic year, we are introducing a strict incomplete policy for the course. The instructor will allow only students with roughly 30% of the course requirements remaining to opt for the INC grade at the end of the semester. The exact 30% requirement can be determined by discussing it with the instructor. For example, if there are 6 assignments and the student completed 4 full assignments. Similarly, if there are 8 labs, the student has 2-3 required labs that are not checked off. If a student chooses the INC option, the remaining work must be completed within 1 month of the end of the course.

For students with > 30% incomplete course requirements, we encourage them to drop the course and retake it next semester. We will not be able to provide the student with an INC grade, and they will automatically get a low grade or NC if they remain in the course. However, by retaking the course, students have the option to transfer their grades on the completed assignments to the next semester. Please let the instructor know that you plan to use this option.

Finally, not completing the course project will count as > 30% incomplete course requirement, and you will not have the option to use an INC. This is because a course project is a whole semester team effort and cannot be completed within the 1-month frame by an individual student for INC completion.

For students planning to graduate in the subsequent semester and have > 30% incomplete course requirements, please discuss this situation with an academic dean and the instructor.

Acknowledgments

Thanks to Tom Doeppner and Laura Dobler for the text on accommodation, mental health, and incomplete policy. Thanks to Karianne Bergen for the text on the policy for AI-generation tools usage. This document was adapted from earlier versions written by Ritambhara Singh, Daniel Ritchie, and Chen Sun.