PSYC 133: Models of Memory

Meeting times	Th 2–3:30; F 3–4:30
Classroom	Moore Library
Instructor	Dr. Jeremy R. Manning
Email	jeremy@dartmouth.edu
Office location	349 Moore Hall
Office hours	By appointment

Course Description

Knowing how our brains organize and spontaneously retrieve memories is at the heart of understanding the basis of the ongoing internal dialog of our conscious thoughts. Put simply, our memories make us who we are. In this course, we will use readings, discussions, and hands-on demonstrations to explore historical approaches, current approaches, and hints about the "next generation" of computational (mathematical) models of learning and memory.

Course Goals

This course is intended to train students to:

- Explore a variety of approaches to building memory models, including neural network models, cognitive process models, and biologically inspired models. Students will implement and train these models, largely from scratch, to gain a deep understanding of how they work.
- Understand different classic and modern approaches to characterizing memory (with a focus on human memory). Students will read primary sources and explore real and synthetic datasets.
- Critically evaluate computational models, construct appropriate tests, and choose appropriate datasets for testing

Pre-Requisites

Students must have prior experience with Python programming in order to do well in this course. Prior coursework on statistics or probability is also highly recommended. Additional prior coursework and/or experience with linguistics, linear algebra, statistics, machine learning, artificial intelligence, data science, philosophy of mind, and/or cognitive models will all be useful, but are not required.

Course Materials

We will use a variety of freely available online materials and research papers, which will be provided throughout the course. You will also need an internet-enabled computer or tablet capable of displaying and outputting graphics and running a standard web browser (e.g., Chrome or Firefox).

Format and Overview

This course follows an experiential learning model. Students will engage with lecture materials through hands-on programming exercises, experiments with models and tools, and group discussions. Problem sets will deepen your understanding of the course material, and small projects will allow you to apply concepts to real-world research problems in the relevant domains.

Classes will include:

- Discussions: Discussions on foundational topics, understanding and implementing models, experiments, and datasets. Class discussions will be recorded, transcribed, anonymized, summarized, and shared with the group. (Note: if you want to bring something up in class that you do not want included in the summary, please let me know either in the moment so that I can pause the recording, or after class so that I can manually scrub it from the final summary.)
- Hands-on Labs: Experimenting with data, designing simple models, and conducting research using these models. All demos will run in Google Colaboratory notebooks.
- Problem Sets and Projects: Bi-weekly problem sets to apply what you've learned. These will typically take the form of small-scale "research" projects where you implement a computational memory model and analyze its behavior using real or synthetic data.

Platforms and Tools

- Google Colaboratory: For developing and running Python code. You may use other programming languages if you prefer, but all in-class coding will be done in Python using Colaboratory.
- GitHub: Used for managing and sharing code, data, and project work.
- Discord: To facilitate discussions outside of class, share ideas, and collaborate on projects.

Grading

Grades will be based on the following components:

- Problem Sets (60%): A total of 4 problem sets designed to reinforce key concepts (each is worth 15% of the final course grade).
- Final Project (40%): You will carry out a larger scale (relative to the problem sets) "research" project on a topic of your choosing. This will include:
  - Python code, organized as a Colaboratory notebook. (Or equivalent if you use another language.)
  - A "presentation" to the class, along with an in-class discussion of your project
  - A brief (2–5 page) writeup of the main approach and key findings or takeaways

Students may work together on any of the assignments, unless otherwise noted in class or in the assignment instructions. However, each student must submit their own problem set and indicate who they worked with. Final projects will (typically) be completed in groups of 2–3 students, with the entire group turning in the same project (and receiving the same grade for it).

Graduate Grading: If you are a graduate student enrolled in the course, you will very likely receive a "P" (Pass), assuming you come to (and participate in) class (with rare exceptions for emergencies or planned absences) and complete all of the assignments. In rare cases, and at the discretion of the instructor, you will earn an "HP" (High Pass) if your work (or participation) goes substantially "above and beyond" what is expected. Conversely, you will earn a "LP" (Low Pass) or "NC" (No Credit) if you do not come to most classes and/or do not complete all of the assignments. My approach to graduate grading is that the onus is on you as the student to decide how much you want to get out of the course. If you put in a lot of effort and work hard, you will learn more. If you seek specific feedback, I will provide it. However, it is certainly possible to pass this course as a graduate student with only a modest effort.

Undergraduate Grading: If you are an undergraduate student enrolled in the course, my grading policy is to simply assign you a letter grade based on the numerical scores you receive on your assignments, weighted as described above. The numbers in parentheses reflect percentages of the total number of possible points: A (93–100), A- (90–92), B+ (87–89), B (83–86), B- (80–82), C+ (77–79), C (73–76), C- (70–72), D (60–69), E (0–59). All grades will be rounded to the nearest integer (e.g., a 92.5 average will result in a final grade of "A", whereas a 92.4999 average will result in a final grade of "A-"). Out of fairness to all students in the course, there will be no "negotiations" about grading– e.g., your grade will be determined solely by your numerical score. Grading on individual assignments will be assigned as follows (for undergraduate students only):

- A: well above and beyond what is expected for the assignment, all components done correctly, well-organized code, clear documentation and explanations, easy-to-read figures, and so on.
- B: default grade, denoting "solid" work– demonstrates clear understanding of the material and meets all expectations for the assignment, with perhaps a few minor errors or other issues.
- C: mostly correct but with a few substantive conceptual or implementation errors.
- D: assignment contains major errors and/or missing components.
- E: either did not turn in the assignment, or did not complete any substantitive part of the assignment.

Detailed feedback will not typically be provided on individual submissions, but you will have the opportunity to raise any issues or questions in class or during my office hours.

Late Policy

For graduate students, problem sets must be submitted prior to the end of the term. To receive specific feedback on the assignment, you must submit it on time (otherwise I will simply mark it as "complete" or "incomplete").

For undergraduates, problem sets will receive a 10% deduction for each week late, rounded up to the nearest whole week (e.g., from a grading standpoint submitting an assignment 1 minute late is the same as submitting it 1 day late, is the same as submitting it 6 days late).

For both graduate and undergraduate students, your final project must be submitted on time (by 11:59PM on the last day of class) in order to receive credit for it. This will ensure that I have time to compile your grades before I need to send them to the registrar.

I strongly encourage you to submit your assignments before the last possible moment to avoid grading penalties, unexpected circumstances (e.g., illness, emergencies, etc.).

Academic Honor Principle

Students are expected to adhere to Dartmouth's Academic Honor Principle. You are encouraged to collaborate and discuss ideas with classmates, but all submitted work must be your own (aside from the final projects, which will be completed in small groups). If you're unsure about what constitutes a violation, please ask for clarification.

Use of Generative AI

The point of this course is for you to learn the material, not to show that you can prompt an LLM to complete your assignments on your behalf. Therefore my strong recommendation (if you actually want to learn this stuff!) is to not use LLMs for your assignments. However, I won't police your work or even attempt to guess whether or not you secretly used Generative AI. Therefore there are just two official "rules" about using Generative AI:

- Most importantly, you are responsible for the content of your assignments, whether written by you "from scratch" or with the help of an LLM (or similar).
- Second, you are bound by the Academic Honor Principle to acknowledge any use of AI in your work. This can be done by either using a brief comment in your code, by explicitly citing the tools you use, or by adding a note to the relevant section(s) of your assignment. Each situation is unique, but you need to make it clear exactly what work is your own vs. produced by AI. You must also include a chat history (including any prompts you used) as an addendum to your assignment(s).

Scheduling Conflicts

Attendance is expected for all classes unless previously arranged. A critical part of the course is the in-class discussions and demos, and those will only work if you are physically present in class. If you anticipate any conflicts due to religious observances or other commitments, please inform the instructor by Week 2 to make appropriate arrangements.

Student Needs

We strive to create an inclusive learning environment where all students feel supported and engaged. If you require any accommodations, please contact the Student Accessibility Services office, or discuss your needs with the instructor privately.