

Computational Neuroscience

Psychology 5270 Spring 2026

January 2, 2026

Course Information

- Class times: TuTh 9:30-11:45
- Class location: Dell 2 100
- Course websites:
 - [github/comp-neurosci](#)
 - [canvas](#)
- Slack group: uva-comp-neuro
 - [Signup](#)
- Instructor: C Daniel Meliza (cdm8j)
 - Office Hours: W 9-10, F 10-11, Gilmer 481
- Teaching Assistant: Ahmad Elsayed (zmk5re)
 - Office Hours: TBD
- Final: in-class presentation

Course Description

To survive and reproduce in a complex and variable physical world, animals have to extract information from sensory inputs, make decisions about how to allocate their resources, and perform tightly choreographed sequences of motor behaviors. Brains have evolved to perform these tasks reliably, flexibly, and efficiently. **Our goal is to understand how these computations are implemented by circuits of neurons.**

The challenge we face is that brains themselves are enormously complex. Even relatively “simple” non-human animals have millions of neurons and billions of connections. Just observing a small fraction of the brain in action can generate enormous quantities of data. Voltage from an intracellular electrode is sampled 40,000 times a second, and EEGs are recorded on multiple channels sampled up to 2,000 times a second. That means almost 2.5 million data points in one minute! With so much data, how can we possibly find the information in all the noise?

The approach we’ll use here is **computational modeling**, which means building toy systems that capture some aspect of how a particular system in the brain is performing its function. As we’ll discuss, there are many different kinds of models and ways of building them, but one of the most useful functions of a model is that it can be used like a lens that can reveal structure in our data and help us filter out the noise.

This course will introduce you to foundational principles of computational modeling in the brain using a hands-on approach with exercises that will also build your expertise in using scientific programming to implement models and apply them to real-world data.

Course Objectives

By the end of the semester, you will be able to:

- identify different types of neuroscience data and the appropriate methods for analyzing them
- explain the physiology of single neurons and networks using dynamical systems theory

- use feedforward and recurrent models to analyze and predict neural activity
- use Python and its numerical libraries to process, analyze, and visualize neuroscience data
- think through computational problems logically and outline the steps of an analysis
- work as part of a team to write and talk about computational research

Course Schedule

This schedule is subject to change based on the interests of the class and our progress through the material, and some readings have yet to be chosen. If there's something you want to cover that isn't listed, let me know! Readings must be completed before class. Lab work must be completed within one week. Non-lab assignment due dates are listed on Canvas.

Date	Activity	Pre-class Readings	Assignment
Foundations			
1/13	Lecture: What is computation?		Pre-course survey
1/15	Lab: Pen and Paper Networks	McNaughton & Morris (1987)	
1/20	Lecture: Dynamical systems	Izhikevich Ch 3–4	Geometrical Analysis
1/22	Lecture: Neuron biophysics	Sterratt Ch 2	
1/27	Lecture: Phenomenological models	Izhikevich Ch 5–6	
1/29	Lab: Single-neuron Simulations	Stimberg et al (2019)	
2/3	Lecture: Point processes	Dayan and Abbott Ch 1	
2/5	Lab: Time-varying Data		
Sensory Systems			
2/10	Lecture: Linear time-invariant systems	Dayan and Abbott Ch 2	Literature Review
2/12	Lab: Linear Systems and Convolution		
2/17	Lecture: Feedforward networks	Ewert (1974)	
2/19	Lab: Receptive Field Estimation		
2/11	Lecture: Decoding sensory activity	Holdgraf et al (2017)	
2/18	Lab: Decoding Models		
2/24	Project journal club		Data Wrangling
2/26	Project journal club		
3/3	(spring break)		
3/5	(spring break)		
Sensorimotor Systems			
3/10	Lecture: Recurrent networks	Dayan and Abbott Ch 7	Hypothesis and Design
3/12	Lab: Network Model Simulations		
3/17	Lecture: Dimensional reduction	Cunningham and Yu (2014)	
3/19	Lab: Manifolds and latent dynamics		
3/24	Lecture: Computation through dynamics	Mante et al (2013)	
3/26	Lab: TBD		
Learning and Control			
3/31	Lecture: Synaptic plasticity	TBD	Preliminary Figures
4/2	Lab: Plasticity Simulation		
4/7	Lecture: Unsupervised learning	TBD	
4/9	Lab: TBD		
4/14	Lecture: Reinforcement learning	TBD	
4/16	Lab: TBD		

Date	Activity	Pre-class Readings	Assignment
4/21	Final presentations		
4/23	Final presentations		
4/26	Final presentations		

Materials

Computing

Scientific programming is essential to computational neuroscience. Digital computers are good at doing the same thing over and over again, which is exactly what we need to reliably deal with the big data generated in neuroscience. In fact, everyone who works with data can benefit from being able to hand tedious tasks off to a computer. Scientific programming is an important skill for researchers at all levels, and the practice of programming will help you learn how to break down complex problems of all kinds into a set of logical steps. Although some assignments will be completed on paper, laptops are necessary for this class. You will need to bring your laptop to every class meeting. Almost all the work will be performed in [Jupyter](#) notebooks hosted on UVA's high performance computing cluster. We will spend time in class making sure everyone has a functioning environment.

Readings

The assigned texts will come from the primary literature and from the following textbooks:

- Dayan, P and Abbott, LF (2001) Theoretical Neuroscience. MIT Press
- Sterratt D, Graham B, Gillies A, and Willshaw D (2011). Principles of Computational Modelling in Neuroscience. Cambridge University Press.
- Eugene Izhikevich, Dynamical Systems in Neuroscience. MIT Press.

The textbooks can be accessed through the UVA library as electronic texts. Use [VIRGO](#) to search. PDF copies will be shared through the File tab in Canvas. Additional readings including brief explainers for probability and linear algebra can also be found on Canvas.

Evaluation

Engagement (40%)

Due to the fact that we complete much of the work in class in pairs and small teams, attendance and active participation are critical Unless otherwise arranged with the instructor, you may only miss one class without penalty. If you are contagious but well enough to attend class virtually, notify the instructor so that you can attend via zoom. Additional absences may be excused due to religious holidays, UVA-required extracurricular activities (e.g., competitions or performances), or legitimate academic reasons; requests must be made and approved at least one week in advance.

How you can succeed: Come to class with a laptop. Read or watch out-of-class material and come prepared with questions and thoughts so you can participate in discussions. Talk to your programming team to make sure you both understand the tasks. Ask questions in class or on the class slack if anything is unclear or if you are getting an error you and your team can't figure out.

Homework (30%)

Each week, you will be given assignments that will immerse you in theoretical or practical aspects of computational neuroscience. Almost all of these assignments will involve programming. Program-

ming isn't just about making sure you've closed all your brackets; it's a new way of thinking, and these problems will give you practice with this type of thinking without your console giving you errors.

How you can succeed: Think hard about the problem and how to construct a series of explicit, logical steps that solve it. Write down your best thoughts on the solution, and read through it carefully to catch any leaps of reasoning you've made. Your process is more important than whether every step is correct. Remember, there are usually multiple ways to solve any problem.

Data Exploration (30%)

As part of a semester-long project, you and your programming team will have the opportunity to choose a neuroscience data set to analyze using one or more of the theoretical frameworks we cover in class. Along the way, you and your team will gain expertise on your data set and be a resource for other students in our class interested in working with that type of data. There will be milestones and chances for revision over the course of the semester. Important dates and more detailed information are laid out in the project description.

How you can succeed: Engage with the data set you have chosen. Explore the structure of your data set and read the methods section of the associated publication. Think about which methods we discuss can be applied to your data and what kinds of questions could be asked with them. Test and document functions you write to make them as easy as possible for you and the rest of the class to use. Take advantage of peer and instructor feedback to improve your final product.

Course Policies

Work must be turned in on time to receive full credit. There is a penalty of 5% per day for late work unless an extension is arranged **before** the due date or an emergency prevents submission. When requesting an extension, provide a brief reason for the request and an estimate of when the work will be completed.

Group work must represent equal effort by all members. Do not put anyone's name on an assignment unless they made a substantive contribution. If your schedules are not compatible, if there is an emergency or illness, or if you need to come to a different arrangement for splitting the work, contact me so we can make sure there is an equitable solution that maximizes everyone's opportunity to learn.

You may not use LLM-generated code for homework assignments. There are many reasons why using LLMs is not ethical (e.g., subsidizing corporate theft of intellectual property and wasteful destruction of natural resources), but the main problem pedagogically is you can only learn how to *do* something through trial and error. One of the most important learning goals in this course is for you to be able to translate models and ideas into code, and you can only get there by building up your mental models and skills piece by piece.

I don't typically answer email or Slack messages outside of normal working hours. If I do write, I don't expect an immediate response.

My goal is for everyone in the class to have an equal opportunity to learn and to demonstrate their knowledge. Students with disabilities are entitled to reasonable accommodations. Contact the Student Disability Access Center (434-243-5180) for more information or to arrange accommodations.

Additional Resources

Short guides to probability and linear algebra can be found on Canvas or the github repo.

Scientific programming and Python

- If you're new to programming in general, check out the [Python for Data Science](#) tutorial.
- VanderPlas J (2016) Python Data Science Handbook. O'Reilly. The full text of this book is available [here](#) as a complete set of Jupyter notebooks that you can run locally or in [Google Colab](#). The first four chapters are highly recommended.
- Langtangen HP (2016) A Primer on Scientific Programming with Python. 5th ed. Springer: Berlin Heidelberg.

Dynamical systems

- Strogatz, *Nonlinear Dynamics and Chaos*. 2nd ed. Taylor and Francis (also see the author's [lectures on youtube](#)).
- Hirsch, Smale, and Devaney, *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Springer.