NSC3270/5270 Computational Neuroscience
Tue/Thu 9:35-10:50am
Featheringill Hall 129

Instructors

Thomas Palmeri
Department of Psychology
507 Wilson Hall
thomas.j.palmeri@vanderbilt.edu
Office Hours: see Brightspace

Sean Polyn
Department of Psychology
504 Wilson Hall
sean.polyn@vanderbilt.edu
Office Hours: see Brightspace

Course Overview

Computational neuroscience is an approach to understanding the information content of neural signals by modeling the nervous system at many different structural scales, including the biophysical, the circuit, and the systems levels.

- T. Sejnowski and T. Poggio

Computational neuroscience is the study of brain function in terms of the information processing properties of the structures that make up the nervous system. It is an interdisciplinary science that links the diverse fields of neuroscience, cognitive science, and psychology with electrical engineering, computer science, mathematics, and physics.

- Wikipedia

In this course, we will use computational techniques to understand how the brain works. We will discuss developing computer simulation models of networks of neurons in the brain; while we will discuss various ways that neurons are computationally instantiated in these models, detailed biophysical and biochemical models of neurons and their cellular and molecular mechanisms will not be covered. Instead, we will discuss different kinds of neural network architectures, different ways of simulating neural dynamics, and different assumptions about how neural networks learn. These models will be used to simulate fundamental aspects of human cognition, including perception, memory, and decision making. We will also discuss using computational approaches to analyze and understand brain data from fMRI, electrophysiology, or neurophysiology. This will include applying psychological, economic, statistical, or machine learning models to brain data. When appropriate, we may discuss how the models and approaches used in computational neuroscience are applied to domains outside of neuroscience. We will rely heavily on demonstrations and hands on experience – in the form of homework assignments – developing, testing, and evaluating computational models and computational approaches.

Prerequisites

For undergraduates in NSC3270, Introductory Neuroscience (NSC2201) or an equivalent course is a prerequisite for enrolling in this course. For graduate students in NSC5270, some knowledge of neuroscience is assumed.

Because most computational models and computational approaches are described in the language of mathematics, and because many of the models we describe are dynamic, involving changes in activity over time, we will assume that students have a basic understanding of calculus, especially the fundamental concepts of the derivative and the integral. For undergraduates, at least one semester of calculus is required. For graduate students without this prerequisite knowledge of elementary calculus, gaps will need to be filled with outside readings and/or videos.

Some knowledge of computer programming in a high-level language (e.g., Matlab, Python, R, C/C++, Java) is required. We will be using Python in this course. We chose Python because of the availability of tools (Keras and Tensorflow) for simulating complex neural network models. For undergraduates, at least one semester of computer programming is required, not necessarily in Python. For graduate students, prior classwork or other experience using some computer programming language, not necessarily Python, is assumed.

We understand that not everyone may know Python. We will quickly cover the basic elements of Python in the first couple of days of class, and we will include sample Python code throughout the semester, but students without knowledge of Python will need to supplement with readings and/or videos, some of which we will suggest.

Laptops

Students are encouraged to bring laptops to class. We may distribute example code before class that will be used during class. We must insist that everyone please refrain from using their laptops for any non-class purposes during class, as it can be very distracting to others.

Course Requirements and Grading

Homework assignments will be handed out regularly – roughly weekly – throughout the course to allow students the opportunity to put the ideas discussed during class into practice. Homework assignments are the primary determinant of the course grade. There will be no graded quizzes or exams. Poor attendance and participation can be used to lower the final grade.

Final letter grades will based on percentages as follows:

		A	92.5 – 100%	A-	90.0 – 92.5%
B+	87.5 – 90.0%	B	82.5 – 87.5%	B-	80.0 – 82.5%
C+	77.5 – 80.0%	C	72.5 – 77.5%	C-	70.0 – 72.5%
D+	67.5 – 70.0%	D	62.5 – 67.5%	D-	60.0 – 62.5%
		F	0.0 – 60.0%		

While we encourage students to help each other out with conceptual misunderstandings, all homework assignments must be completed individually. Unexcused late assignments will be penalized 10% for every 24 hours late, starting from the time class ends, for a maximum of two days, after which they will earn a 0.

Any student auditing the course is expected to attend class and can participate in a way commensurate with the amount of work they do on class homework assignments.

Grader

The grader for this course is Andrew Bender. If you have questions about how your homework was graded, they can be addressed to him at andrew.s.bender@vanderbilt.edu. Andrew is not a resource for help with the homework, which should be addressed to the professors.

Graduate Student Requirements

Graduate students are required to enroll in NSC5270. In addition to completing the homework assignments and attending class, graduate students are required to complete a small final project in Python that relates somehow to the principles and approaches discussed in class. Graduate students will meet with one of the professors to discuss potential projects and get our okay to proceed. The final submitted project will consist of Python code that can be run and a short, one or two page, description of what the code is used for. The final project can be closely related to your graduate research but it does not need to be. The final project will constitute 10% of your final course grade.

Brightspace

All course materials (readings, web links, lecture slides, homework assignments, example code) will be posted on the Brightspace site for this course. You will also turn in your homework assignments on Brightspace; while you can turn in a homework assignment more than once, we will only look at and grade the last version you turn in. You can also view your grades and comments for the assignments within Brightspace.

Readings

There is no book for this course. Instead, course readings (book chapters, articles, links to videos and web sites) will be posted on Brightspace and will be announced in class.

Python

The Python programming language will be used for all assignments in this course.

Python is a high-level computer programming language particularly well suited to computational neuroscience applications. It is free, open software that runs on multiple platforms (Windows, Mac, and Linux). It is highly extensible with thousands of libraries and modules written and shared by scientists and engineers from around the world. It allows easy interface with programs written in languages like C, Fortran, Java, or Matlab.

We will be using the Anaconda Python 3.6 distribution and associated libraries (particularly numpy, scipy, and matplotlib) in this class. We are recommending that students download and use the free Wingware Python IDE. We will also use the Keras and Tensorflow modules. Details on how to download and install these are provided in a document uploaded to Brightspace.

Python Resources

There is no textbook for this course. Two suggested Python references are:

A Whirlwind Tour of Python by Jake VanderPlas, available as a free ebook at
http://www.oreilly.com/programming/free/a-whirlwind-tour-of-python.csp
with associated Jupyter notebooks (code you can run) at
https://github.com/jakevdp/WhirlwindTourOfPython

Also potentially useful is the *Python Data Science Handbook* by Jake VanderPlas, available at
http://shop.oreilly.com/product/0636920034919.do
and also in the form of free Jupyter notebooks at
https://github.com/jakevdp/PythonDataScienceHandbook

Other readings and videos will be posted on Brightspace.

Tentative Course Topics

The complete schedule of topics will be outlined and updated regularly on Brightspace. Here is a tentative list of topics and approaches we may discuss over the course of the semester:

Introduction to Computational Neuroscience
Introduction to Python
Simple Feedforward Networks
Perceptrons
Multi-Layered Neural Networks
Supervised Network Learning
Unsupervised Network Learning
Hopfield Networks
Recurrent Neural Networks
Dynamic Neural Networks
Reinforcement Learning
Integrate and Fire Networks
Spiking Neural Networks
Deep Learning and Convolutional Neural Networks
Neural Models of Visual Perception
Neural Models of Memory
Neural Models of Decision Making
Scalp Electroencephalography (EEG) Analyses
Intracranial Electrocorticography (ECoG) Analyses
Analyzing Neural Oscillations
Functional Magnetic Resonance Imaging Analyses
Multivariate Pattern Analysis Techniques
Representational Similarity Analysis
Model-based Cognitive Neuroscience

Vanderbilt's Honor Code Governs All Work in this Course