Computational Cognitive Modeling

Brenden Lake & Todd Gureckis

Brenden Lake

Assistant Professor, Data Science and Psychology
Research Scientist, Facebook AI Research

office hours: Wednesdays
1-2pm, zoom

https://cims.nyu.edu/~brenden
https://lake-lab.github.io/

Todd Gureckis

Associate Professor, Psychology
Affiliate, Center for Data Science

office hours: Tuesday 11am-12 and Wednesday 4-5pm; zoom

http://gureckislab.org

Guy Davidson

2nd year PhD student, Data Science

office hours: Tuesdays 1-2pm, zoom

Aysja Johnson

2nd year PhD student, Psychology

office hours: Mondays 3-4pm, zoom


LOGISTICS

Course website

https://brendenlake.github.io/CCM-site/

Computational cognitive modeling - Spring 2021

NYU PSYCH-GA 3405.004 / DS-GA 1016.003

View On GitHub

This project is maintained by brendenlake

Hosted on GitHub Pages using the Dinky theme

Computational cognitive modeling - Spring 2021

Instructors: Brenden Lake and Todd Gureckis

Teaching Assistants: Guy Davidson and Aysja Johnson

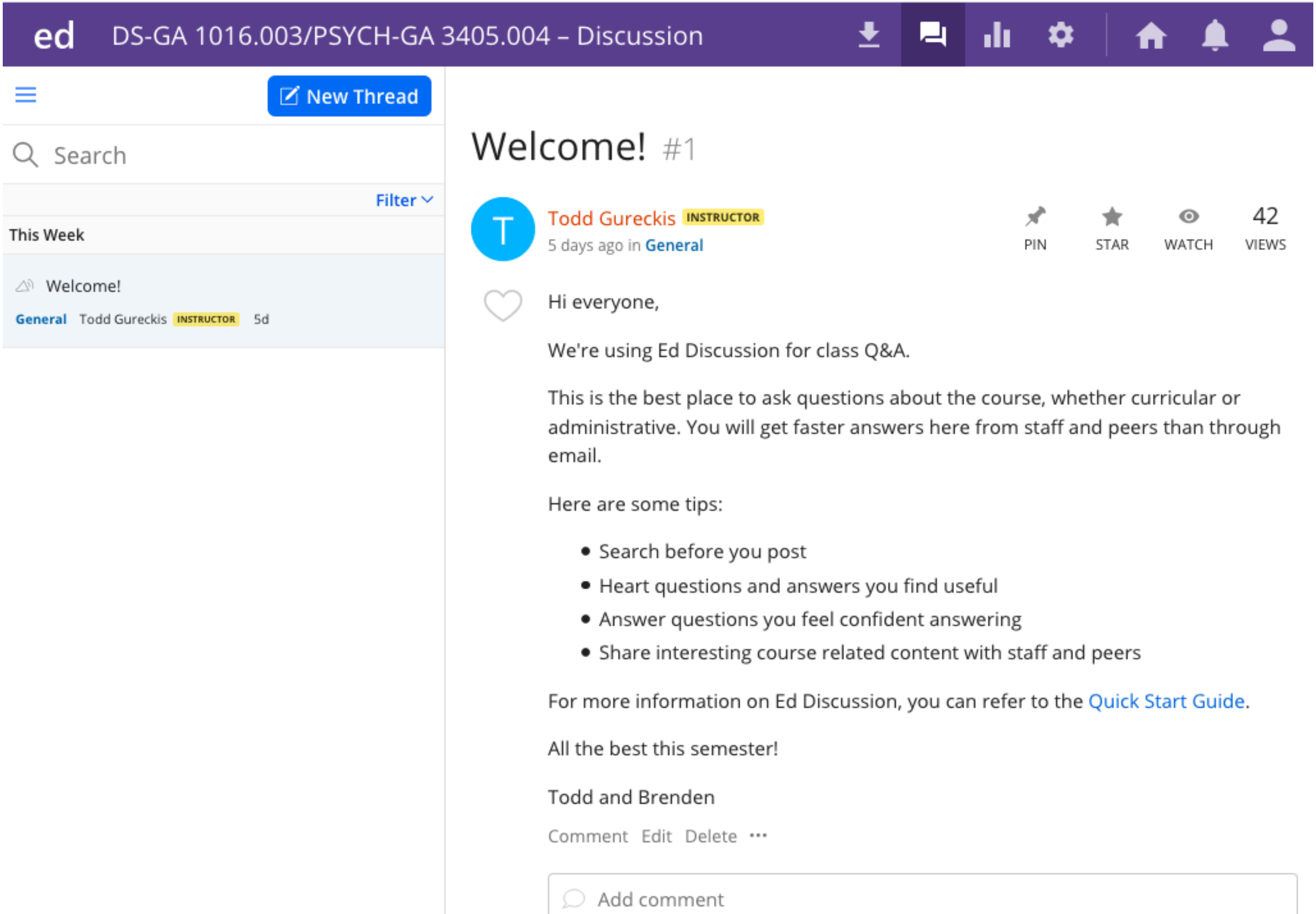
Meeting time and location:
Lecture. We have a "flipped classroom" model. Lectures will be pre-recorded and available to watch on Vimeo. Please watch the lecture *before* the date it is listed under, so that we have a productive live discussion. Please come ready with your questions.
Live discussion of the lectures is blended on Mondays 1:35-2:35 PM.
In person GCASL_C95 - Global Center for Academic & Spiritual Life, 238 Thompson Street (capacity = 56).
Zoom link in NYU Classes.

Labs. We have two times. Attend only one, whichever is more convenient for you.
Option 1: Tuesdays 8:00-8:50 AM
Online, Zoom link in NYU Classes.

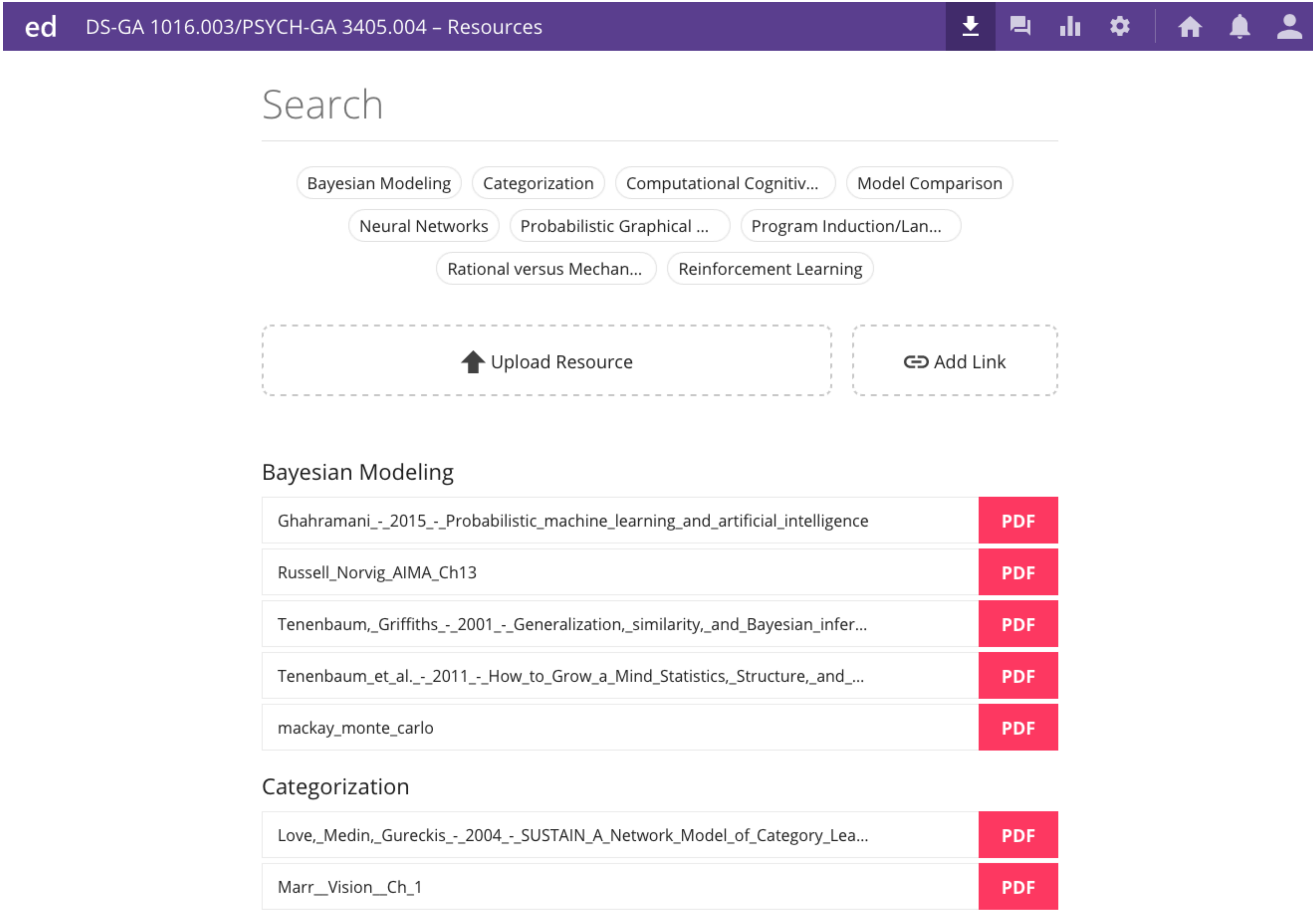
Option 2: Tuesdays 2:40-3:30 PM
Blended (60FA, capacity 17 students) also can access via the Zoom link in NYU Classes.
(**Labs are remote only for the first two weeks.**)

Course numbers:
DS-GA 1016.003 (Data Science)

Course discussion: EdStem



Readings posted on EdStem



Getting in touch

EdStem should be your main point of contact. If you have a question, and you think there is a possibility that someone may have the same question, please post it to EdStem for everyone's benefit.

If you need to send an individual message,

Email address for instructors and TAs:
instructors-ccm-spring2021@nyuccl.org

Flipped classroom

Pre-recorded lectures on Vimeo

(Links on course website) Please watch before the date they are listed, so we can have a robust discussion!

Blended discussion of lecture:

Mondays 1:35-2:35 PM

Zoom (https://nyu.zoom.us/j/92201764817)
OR
Global Center for Academic & Spiritual Life,
Room C95

Labs

We have two times.

Option 1: Tuesdays 8:00-8:50 AM

Online only, Zoom link: https://nyu.zoom.us/j/94450481642

Option 2: Tuesdays 2:40-3:30 PM

Blended (60 Fifth Avenue, Room 150, capacity 17 students; Rotates between cohort B and C)

OR Zoom link: https://nyu.zoom.us/j/98468890846

Lecture schedule

Mon Feb 01: Introduction (video)(slides-intro)
Mon Feb 08: Neural networks / Deep learning (part 1)
Mon Feb 15: No class, President’s Day
Thursday Feb 18 (Legislative Day, with Monday schedule): Neural networks / Deep learning (part 2)
Mon Feb 22: Reinforcement learning (part 1)
Mon Mar 01: Reinforcement learning (part 2)
Mon Mar 08: Reinforcement learning (part 3)
Mon Mar 15: Bayesian modeling (part 1)
Mon Mar 22: Bayesian modeling (part 2)
Mon Mar 29: Model comparison and fitting, tricks of the trade
Mon Apr 05: Categorization
Mon Apr 12: Probabilistic Graphical models
Mon Apr 19: No class, Long Weekend (Spring Break Replacement)
Mon Apr 26: Information sampling and active learning
Mon May 03: Program induction and language of thought models
Mon May 10: Computational Cognitive Neuroscience

Lab schedule

Tue Feb 02, Python and Jupyter notebooks review (**remote only this week**)
Tue Feb 09, Introduction to PyTorch (**remote only this week**)
Tue Feb 16, HW 1 Review
Tue Feb 23, TBD
Tue Mar 02, TBD
Tue Mar 09, HW 2 Review
Tue Mar 16, Probability Review
Tue Mar 23, HW 3 Review
Tue Mar 30, TBD
Tue Apr 06, TBD
Tue Apr 13, HW 4 Review
Tue Apr 20, TBD
Tue Apr 27, TBD
Tue May 04, TBD

Pre-requisites

- *Math:* We will use concepts from linear algebra, calculus, and probability. If you had linear algebra and calculus as an undergrad, or if you have taken Math Tools in the psychology department, you will be in a good position for approaching the material. Familiarity with probability is also assumed. We will review some of the basic technical concepts in lab.

- *Programming:* Previous experience with Python is required. Previous IN CLASS experience with Python is strongly recommend—it's assumed you know how to program in Python. The assignments will use Python 3 and Jupyter Notebooks (http://jupyter.org)

Grading:

- The final grade is based on the homeworks (60%) and the final project (40%).

Final project:

- The final project will be done in groups of 3-4 students. A short paper will be turned in describing the project (approximately 6 pages). The project will represent either an substantial extension of one of the homeworks (e.g., exploring some new aspect of one of the assignments), implementing and extending an existing cognitive modeling paper, or a cognitive modeling project related to your research. We provide a list of project ideas (see website), but of course you do not have to choose from this list.

Homeworks — programming requirements

Programming: We assume you are familiar with programming in Python

Homeworks use this setup:

- Python 3
- Jupyter notebooks
- Standard Python packages for scientific computing
  - numpy
  - scipy
  - pandas
  - matplotlib
- PyTorch >=1.4 library for neural networks

Using your laptop setup is encouraged!

Jupyter notebooks

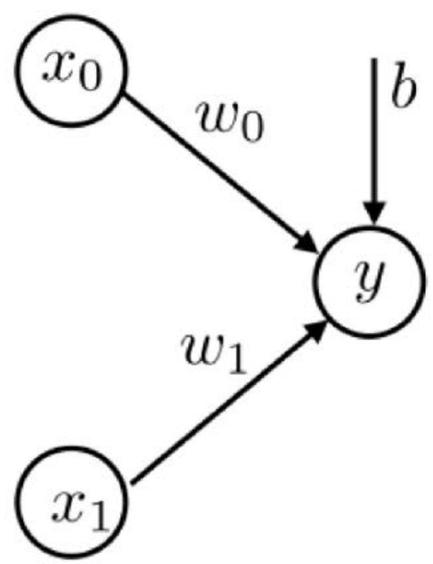
Homework - Neural networks - Part B (20 points)

Gradient descent for an artifical neuron

by *Brenden Lake* and *Todd Gureckis*
Computational Cognitive Modeling
NYU class webpage: https://brendenlake.github.io/CCM-site/
email to course instructors: instructors-ccm-spring2019@nyuccl.org

This homework is due before midnight on Monday, Feb. 25, 2019.

This assignment implements the gradient descent algorithm for a simple artificial neuron. As covered in lecture, the neuron will learn to compute logical OR. The neuron model and logical OR are shown below, for inputs x_0 and x_1 and target output y.



logical OR

x_0	x_1	y
0	0	0
0	1	1
1	0	1
1	1	1

This assignment requires some basic PyTorch skills, which were covered in lab. You can also review two basic [PyTorch tutorials](), "What is PyTorch?" and "Autograd", which have the basics you need.

In [ ]:
```
# Import libraries
from __future__ import print_function
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import torch
import torch.nn as nn
```

Let's create `torch.tensor` objects for representing the data matrix `D` with targets `Y`. Each row of `D` is a different data point.

In [ ]:
```
# Data
D = np.zeros((4,2),dtype=float)
D[0,:] = [0.,0.]
D[1,:] = [0.,1.]
D[2,:] = [1.,0.]
D[3,:] = [1.,1.]
```

Pre-configured cloud environment

Students registered for the course have the option of completing homework assignments on their personal computers (encouraged if know how to set it up!), or in a cloud Jupyter environment with all required packages pre-installed (see website).

Collaboration and honor code

We take the collaboration policy and academic integrity **very seriously**. Violations of the policy will result in zero points and possible disciplinary referral.

You may discuss the homework assignments with your classmates, but you must run the simulations and complete the write-ups for the homeworks on your own. Under no circumstance should students look at each other's code or write ups, or code/write-ups from previous years of this course. Do not share your write up or code with any of your classmates under any circumstances.

Course policies

Late work:

- We will take off 10% for each day a homework or final project is late.

See policy on extensions, regrading, extra credit, etc. on syllabus

Background survey

- Currently enrolled in what type of program:
  - Psychology Ph.D.? Psychology Masters? Data Science Masters? DS Ph.D.? Other graduate program? Undergraduate?

- Previous coursework:
  - Cognitive Psychology? Programming? Probability, statistics, MathTools? Machine learning? AI? Deep learning?

- Who knows about:
  - Classical conditioning?
  - Prototype vs. exemplar models?
  - Categorical perception?
  - Semantic networks?
  - Logistic regression?
  - Backpropagation algorithm?
  - Simple recurrent network?
  - Model-based vs. model-free reinforcement learning?
  - Bayes' rule?
  - Conditional independence?
  - Conjugate prior?
  - Metropolis-Hastings?
  - Explaining away?
  - Probabilistic programming?

What you will come away with…

1. Experience with the major paradigms for computational cognitive modeling

2. An introduction to key technical tools (in Python and Jupyter notebooks):

- Neural networks / deep learning (in PyTorch)
- Reinforcement learning
- Bayesian modeling
- Model comparison and fitting
- Probabilistic graphical models
- Program induction and language of thought models

3. How to build computational models to test and evaluate psychological theories, and to understand behavioral data by modeling the underlying cognitive processes.

4. Ideally, students will leave the course with a richer understanding of how computational modeling advances cognitive science, and how computational cognitive modeling can inform research in data science, machine learning, and artificial intelligence

Is this course a substitute for machine learning?

- **No. It's not a substitute, it's complementary.**
- This course does survey different computational paradigms (deep learning, reinforcement learning, Bayesian modeling, classification, graphical models, etc.), and there is some overlap with ML classes in terms of technical content.
- But unlike ML classes, this is also a cognitive science class. **Our examples and applications aim to understand human learning, reasoning, and development, and to understand intelligent behavior more generally.**
- We get into some mathematical background, but ML courses take a more formal approach than we do here. We aim for a more accessible introduction.
- You will get hands on experience with running and analyzing complex models, implementing some (but not all) models, and analyzing behavioral data with computational models. Extensive final project.

For next time....

Readings for the next two lectures (available on EdStem)

- McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. The Appeal of Parallel Distributed Processing. Vol I, Ch 1.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. Nature 521:436–44.
- McClelland, J. L., & Rogers, T. T. (2003). The parallel distributed processing approach to semantic cognition. Nature Reviews Neuroscience, 4(4), 310-322.
- Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179-211.
- Peterson, J., Abbott, J., & Griffiths, T. (2016). Adapting Deep Network Features to Capture Psychological Representations. Presented at the 38th Annual Conference of the Cognitive Science Society.

Homework 1 on neural networks will be released next class

Questions?