Syllabus and FAQ
Applied Deep Learning, Spring 2019
COMS W4995.07

Instructor: Josh Gordon
Day / Time: Thursdays, 7pm - 9:30pm
Where: 207 Mathematics Building

Description
This course provides a practical, hands-on introduction to Deep Learning. We aim to help students understand the fundamentals of neural networks (DNNs, CNNs, and RNNs), and prepare students to successfully apply them in practice. This course will be taught using open-source software, including TensorFlow 2.0. In addition to covering the fundamental methods, we will discuss the rapidly developing space of frameworks and applications, including deep learning on the web. This course includes an emphasis on fairness and testing, and teaches best practices with these in mind.

Syllabus

Subject to change a bit.

Date	Topic(s)
1/24	**Introduction** Topics: deep learning overview, recent developments.
1/31	**Linear models and famous incidents** Topics: weights, loss, activation functions, softmax. Data-driven incidents and why they happened.
1/31	Assignment 1 released
2/7	**Working with images** Topics: CNNs, transfer learning, data augmentation, famous models.
2/9	Assignment 2 released
2/14	**Optimization** Topics: gradient descent, backpropagation.
2/14	Assignment 1 due
2/21	**Open source frameworks** Topics: TensorFlow, low and high level APIs, symbolic vs imperative models.

	Assignment 2 due
2/21	Assignment 3 released
2/28	**Case studies in medical imaging** Topics: recent work in depth.
2/28	**Project description posted** Choose from a suggested project or propose your own. Start on your project early!
3/7	**Midterm review** Josh has a conference this week. TAs will hold a review session, covering practice problems similar to the exam.
3/7	Assignment 3 due
3/7	Project proposal due (only if doing a custom project)
3/14	**Deep Learning in Javascript** Topics: interactive, client-side ML on the web.
3/14	Assignment 4 released
3/21	*Reminder: no class (university holiday)*
3/28	Midterm
4/4	**Working with text and timeseries data** Topics**:** RNNs, text classification, text generation, CNNs for non-image data.
4/4	Assignment 4 due
4/11	**Working with structured data; testing** Classifying structured data, interpreting models, testing data-driven applications.
4/11	Assignment 5 released
4/18	**Reinforcement learning; optimization cont'd** Guest lecture on RL, batchnorm, weight initialization strategies.
4/25	**Generative models and advanced topics** GANs, VAEs, NMT, image captioning.
4/25	Assignment 5 due
5/2	Poster session and project submission

FAQ

Prerequisites and difficulty level
Data science students and practitioners come from diverse academic backgrounds. It's important to the instructor this course is accessible.

- Students should be comfortable programming in Python (including NumPy and Matplotlib). The Python Data Science Handbook is a helpful reference (the content is available for free online).

- Although the department requires that students have previously taken COMS W4721 (Machine Learning for Data Science) or an equivalent course, we will review important concepts as we go.

- Assignments will be of intermediate difficulty.

This course is right for you if you're interested in developing practical skills. The department also offers theoretical courses if those match your interests.

Extra credit / challenge problems
You have a few opportunities if you'd like to stretch yourself beyond the requirements (all of these are optional):

- You can propose your own course project if there's a particular area you'd like to explore.

- The suggested course project offers you the chance to train models on about 1TB of data (students can complete it for full credit using only several GBs).

- Some assignments includes extra credit problems (for example, reproduce this visualization). Others are practical, for example, train a model to recognize landmarks on campus, and deploy it on a phone.

Workload and grading

- Five homework assignments (60%). All will be practical (e.g., develop an image classifier).

- A course project and presentation (25%).

- A midterm (15%). At most a couple questions will involve calculus, the rest will be short answer written responses. There is no final exam.

How will extra credit factor into grading?

- It's a deciding factor if you fall in between two grades (e.g., to boost your grade from an A- to an A).

How can I get an A+?

- It's very hard. You will need to do exceptional work on a couple challenge problems (out of >100 students last year, only one received an A+).

Course project
The course project gives you an opportunity to create a portfolio of work. Students may choose to complete a suggested project, or propose their own.

The suggested project involves developing a solution to the CAMELYON16 challenge; starter code and data will be provided. Note that it's relatively easy to develop a basic solution, but difficult to develop a good one.

The project may be done in groups of up to three. If you're doing a group project, the expectations will be higher. Either way, deliverables include:

- A link to your GitHub repo
- A poster session
- An YouTube video (+/- 10 minutes, including a presentation, demo, and code walkthrough). The video can be unlisted, and does not need to be shared publicly.

Textbooks
There are two. Students should purchase a copy of Deep Learning with Python. This is light reading, and full of practical advice. We will supplement this book with Deep Learning by Ian Goodfellow (available for free online).

Software and languages
There several open-source deep learning frameworks available today. We will use TensorFlow's Python APIs, with best practices for TensorFlow 2.0 in mind (coming in mid 2019). Students are welcome to complete assignments using other frameworks[1] if they prefer (e.g., JAX, PyTorch, MXNet, etc). One assignment includes a small amount of JavaScript, prior experience is not assumed.

Programming environment
We recommend Colaboratory, a web-based Jupyter environment that includes a free GPU. Alternatively, students may use university hardware or the cloud platform of their choice.

Late policy

[1] Once you know one framework, it is relatively easy to transition to others.

Students will have two or three weeks to complete each assignment from the day they're assigned (not because we expect them to take a long time(!), rather, so you're free to plan your work as is convenient). After the due date, the penalty for late assignments is 10% / day. After three days, late assignments will no longer be accepted.

Solutions to extra credit problems can be submitted anytime up to the last day of class. After that, they will no longer be accepted.

Class format
We have a long time slot. We will use some class time to work on homework, please bring your laptop.

Collaboration policy
Feel free to study in groups. You may discuss your approach to homework assignments, and help each other debug. That said everyone must write and submit their own code (with the exception of the course project, if you are working in a group). Please keep the university's academic integrity policy in mind.

Exam policy
The midterm is closed-book and closed-notes.

Disability Services
Disability Services facilitates equal access for students with disabilities by coordinating accommodations and support services, cultivating a campus culture that is sensitive and responsive to the needs of students. Students seeking accommodations or support services from Disability Services are required to register with the office. If you are interested in pursuing an evaluation for a learning disability, please visit the referrals and other campus resources page.

Office hours
- Instructor: Mudd 417 from 5:30pm - 6:30pm on Thursdays.
- TAs: TBD.