Course Missive

Fall 2017

Time and Location: MWF 12:00-12:50 PM, Salomon Hall 001

Course Staff

<table>
<thead>
<tr>
<th>What</th>
<th>Who</th>
<th>Where</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor</td>
<td>Eugene Charniak (ec)</td>
<td>CIT 419</td>
<td>Hours by appointment</td>
</tr>
<tr>
<td>Head TA</td>
<td>Sidd Karamcheti (skaramch)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td>Undergraduate TAs</td>
<td>Arun Drelich (adrelich)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Dilip Arumugam (darumuga)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Evan Cater (ecater1)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Jacob Beck (jab11)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Melrose Roderick (maroderi)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Nathaniel Brennan (nbrennan)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Raphael Kargon (rkargon)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Sean Segal (ss97)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Zhenhao Hou (zhou)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td>Raphael Kargon (rkargon)</td>
<td>TBA</td>
<td>TBA</td>
</tr>
</tbody>
</table>

Introduction

Welcome to CS147! Over the past few years, Deep Learning has become a popular area, with deep neural network methods obtaining state-of-the-art results on applications in computer vision (Self-Driving Cars), natural language processing (Google Translate), and reinforcement learning (AlphaGo). This course intends to give students a practical understanding of the field of Deep Learning, through lectures and labs covering both the theory and application of neural networks to the above areas (and more!). We introduce students to the core concepts of Deep Neural Networks, including the backpropagation algorithm for training neural networks, as well as specific operations like convolution (in the context of computer vision), and word embeddings and recurrent neural networks (in the context of natural language processing). We also teach the Tensorflow Framework for the expression of deep neural network models.

Prerequisites

- A basic programming course: (CSCI 0150, 0170 or 0190)
- A linear algebra course: (CSCI 0530, MATH 0520 or 0540)
- A stats / probability course: (CSCI 0220, 1450, 0450, MATH 1610, APMA 1650 or 1655)
Assignments

There are nine programming projects as well as weekly labs. The grade breakdown for this course is:

- 70% – Programming Projects
- 20% – Weekly Labs
- 10% – In-class Participation

Grad students taking this class will also need to do an open-ended final project. This will be 25% of the course grade, and other components will be scaled down accordingly:

- 25% – Final Project
- 53% – Programming Projects
- 15% – Weekly Labs
- 7% – In-class Participation

Note, however, that you must complete all of the programming projects to receive credit for this course.

Each of the programming projects is due by midnight on the given due date — thus, you have until the end of the day listed as the due date to complete the assignment. You will have seven (7) free late days to use towards all but the last project. After your late days are expended you will lose 10% of your project grade for each extra day your handin is delayed. At the end of the term, we will calculate how to best divvy out your late days to best help your final grade. There is no need for you to do anything special for this, other than keep track of submission dates for your own records. Projects handed in after 11:59pm on Tuesday, May 10th will not be accepted.

Labs

Students will be assigned to a weekly lab slot. Students will then have an hour to complete the lab for that week and get it checked off by TAs. If they do not complete the lab / get it checked off in that time slot, they can also come to TA hours to get the lab checked off. However, labs should be checked off by the end of the week in which they were assigned.

Coding

This course will use Python and its Tensorflow API. Projects should all be done in Python, and labs will be provided as iPython notebooks.

Since this is not a software engineering course, we won’t be enforcing stringent style guidelines, but you should write so that someone who isn’t a wizard with your language of choice will be able to understand what your program is doing (add plenty of comments, break up code into smaller
functions, i.e. apply basic common sense). If you turn in a partially-functional assignment and we can’t tell what you were trying to do, we’ll probably be very grumpy about giving partial credit. As that translates to an official policy, so long as your code produces the expected output(s) and adheres to any specific project restrictions (runtime, etc.) then you will not lose points for poor design or coding practices. However, as this is not a software design course, it is not the responsibility of the TAs to attempt to understand the intentions underlying confusing code. If it is not fully clear what you were trying to do in the implementation of a partially-functional assignment (i.e. not all of the output is as expected) then partial-credit will be given sparingly, and at our discretion.

**Time Requirements**

In addition to 3 hours per week in class, you will probably need 1 hour of help from a UTA, 1 hour for lab, 2 hours of reading and review to solidify your grasp of the material, and 6 hours for the programming assignments. (182 hours/semester)

**Extensions / Incompletes**

Requests for extensions on assignments or incompletes for the course should be directed to Professor Charniak. However, please note that Charniak will be very reluctant to grant an extension or incomplete without a dean’s note or other documentation indicating an extenuating circumstance.

**Collaboration Policy**

Discussion of material with your classmates is both permitted and encouraged. However, showing, copying or other sharing of actual code is forbidden. This will be enforced.