CSC321 Winter 2015 - Course information
Introduction to Neural Networks

Course web site: `http://www.cs.toronto.edu/~rgrosse/csc321/`
Coursera lectures, quizzes, and forum: `https://utoronto.coursera.org/CSC321-003`

Overview

Machine learning is a powerful set of techniques that allow computers to learn from data rather than having a human expert program a behavior by hand. Neural networks are a class of machine learning algorithm originally inspired by the brain, but which have recently have seen a lot of success at practical applications. Theyre at the heart of production systems at companies like Google and Facebook for face recognition, speech-to-text, and language understanding.

This course gives an overview of both the foundational ideas and the recent advances in neural net algorithms. Roughly the first 2/3 of the course focuses on supervised learning — training the network to produce a specified behavior when one has lots of labeled examples of that behavior. The last 1/3 focuses on unsupervised learning — the algorithm isn't given any examples of the correct behavior, and the goal is instead to discover interesting regularities in the data.

Format

This course is taught using the "inverted classroom" model. This means that instead of being introduced to the material in a largely one-way lecture in a hall, you'll watch the lecture as a video at home before class, and then in class, we will discuss the material in more depth and work through problems as a class. We will also have a small number of traditional in-class lectures which will introduce new developments since the videos were recorded 3 years ago. (It's a fast-moving field!)

In order to encourage you to think about the videos carefully before lecture, we assign short quizzes, which are **due at 11:59pm on Monday** immediately preceding the class meetings which cover the material. You do the quizzes on the Coursera web site, and they are graded automatically. They're not meant to be an important part of the evaluation, but are primarily a way to get quick feedback on your understanding.

To make the class more interactive, we also have you ask us a question at the end of each quiz. We will try to answer the most common questions, and clear up the main points of confusion, during class time.

Accessing the Coursera materials. The video lectures, quizzes, and forum are all found at the Coursera page for the course (`https://utoronto.coursera.org/CSC321-003`). You must sign in using your UTorID. **If you already have a separate Coursera account, *do not* sign in with that. The publicly accessible course is not the same as what we're using.** You should be able to access the materials even if you're not registered for the course (e.g. if you are on the waiting list). Please let us know if you have any problems accessing the materials.

Schedule

There is both an afternoon section and a night section for the course. Both will cover the same material, and will have the same assignments and exams. Since both sections are at full enrollment, **please attend your assigned section**.

Note that the "lecture" periods will not be lectures in the traditional sense, because of the inverted classroom model that we're using.

Some weeks there is no tutorial. See the calendar page (`http://www.cs.toronto.edu/~rgrosse/csc321/calendar.html`) for details of the tutorials.

- **Afternoon section**
  Lectures: Tuesdays and Thursdays, 1:10 - 2:00 pm, in Bahen, room 1200.
  The first lecture is on January 5; the last lecture is on April 2. There are no lectures on February 17 and 19 (reading week).
  Tutorials: Thursdays 12:10 - 1:00 pm (also in BA 1200).

- **Night section**
  Lectures: Tuesdays, 6:10-7:50 pm, in Bahen, room 1220.
  The first lecture is on January 5; the last lecture is on March 31. There is no lecture on February 17 (reading week).
  Tutorials: Tuesdays, 8:20-9:00 pm (also in BA 1220). This leaves half an hour for a quick dinner between lecture and tutorial.

Teaching team

Instructor (afternoon section)
Roger Grosse
e-mail: `csc321prof@cs.toronto.edu`
Office hours: Mon 2-3pm in Pratt 290C

Instructor (night section)
Nitish Srivastava
e-mail: `csc321prof@cs.toronto.edu`
Office hours: Mon 3-4pm in Pratt 290C

Teaching assistants:
Jimmy Ba
Hannes Bretschneider
Ryan Kiros
Kaustav Kundu
Vishal Raheja
Shikhar Sharma

TA e-mail: `csc321ta@cs.toronto.edu`

Do NOT send us email about the class directly to our personal accounts. We will not answer.

Prerequisites

If you want a waiver, you MUST contact your instructor in the first week. The prerequisites that will be enforced are:

- (MAT135H1, MAT136H1)/MAT135Y1/MAT137Y1/MAT157Y1
- MAT223H1/MAT240H1
- STA247H1/STA255H1/STA257H1
- CGPA 3.0, or enrollment in a CSC subject POSt

If you feel that you have learned the material from another source, we will be generous about approving waivers.

Load

There are 24 hours of lectures and 6 hours of tutorials.

Readings

There is no required textbook for the class. Most of the course material is presented in the lecture videos. A few small readings may be assigned if the need arises. These required readings will all be available on the web, for free.

There are also some relevant resources which are freely available online. We will try to provide links on a lecture-by-lecture basis.

- *Metacademy*, an online website (which one of the instructors is involved with) which helps you construct personalized learning plans and which has links to lots of resources relevant to particular concepts. We'll post links to relevant Metacademy concepts as the course progresses.
  `http://www.metacademy.org`
- *Deep Learning*, a draft textbook by Yoshua Bengio, Ian Goodfellow, and Aaron Courville.
  `http://www.iro.umontreal.ca/~bengioy/dlbook/`
- Video lectures for Hugo Larochelle's neural networks course. These are similar to Geoff's lectures but a bit more mathematical.
  `http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html`
- David MacKay's excellent textbook, *Information Theory, Inference, and Learning Algorithms.* This isn't focused on neural nets per se, but it has some overlap with this course, especially the lectures on Bayesian models.
  `http://www.inference.phy.cam.ac.uk/mackay/itila/`

- *Neural Networks and Deep Learning*, a book by physicist Michael Nielsen which covers the basics of neural nets and backpropagation.
  `http://neuralnetworksanddeeplearning.com/`

Marking Scheme

- Midterm test: 15%.
- Final exam: 35%.
- Four assignments worth 10% each.
- The quizzes on the Coursera website: 10%.

By the time you get to an advanced course like csc321 you've heard this lots of times, so we'll keep it brief: avoid academic offenses (a.k.a. cheating). All graded work in this course is individual work.

Quizzes

10% of your grade is based on a number of quizzes (about 15 of them), which you will complete on the Coursera website. These are not designed to be difficult, or to be an important part of the evaluation. Rather, they are mostly designed to help you figure out which parts of the material you understand and which parts need more work.

You get two attempts for each quiz. The greater of the two scores is recorded. Even if you get a perfect score on your first attempt, you're encouraged to go through the second attempt, because the questions will be slightly different and that will help you think about the material.

Each quiz will be due at 11:59pm Monday night *before* the lectures which cover the corresponding material. Because we use class time to expand on the material in these lectures, it's important that you watch them ahead of time so that you have a basic understanding.

You can open the quiz whenever you want, and then think about the questions until just before the deadline, if you want. If you complete it late, your recorded score for that quiz will be zero. Quiz posting dates and due dates are listed on the course calendar.

Assignments

A typical assignment will require you to write (or modify) and use some Python code that implements a simple version of a learning procedure that has recently been covered in the course. You will have to submit a very brief report (one page plus figures) that describes the results you obtained.

Dates. See the calendar page for assignment due dates. All assignments are due on a Tuesday, **in class, on paper, at the start of your assigned section, i.e. 1:10pm or 6:10pm. Please**

do not hand them in at the other section. If that is somehow impossible, please contact your instructor to explain the situation, and we'll figure something out.

Lateness. Except in the case of an official Student Medical Certificate, assignments that are submitted late will be graded out of 70%, 50%, or 0% of the full score, depending on whether they are 1, 2, or more days late. The time past the deadline will be rounded **UP** to an integer number of days.

Collaboration policy. You are expected to work on the assignments by yourself. You should not discuss them with anyone except the tutors or the instructor. The report you hand in should be entirely your own work and you may be asked to demonstrate how you got any results that you report.

Tests

Midterm. The midterm test (worth 15% of the course grade) will be held during class time, in the same room as usual, on **Tuesday, 2/24**. It is closed book. It covers all material before reading week, *i.e.* up through the material covered on Thursday, 2/12. We will link to practice midterms.

1. The midterm for the afternoon section will be held during lecture time, **1:10-2pm on Tuesday, 2/24**.

2. The midterm for the night section will be held during lecture time, **6:10-7pm on Tuesday, 2/24**.

Final exam. The final exam is worth 35% of the course grade. It is a closed book exam. About 25% of the questions will be based on material that came before the midterm and about 75% on material that came after the midterm.

Missed tests. Missed tests will get a score of 0 except in the case of an official Student Medical Certificate or a **written** (not email) request submitted at least one week before the test date and approved by the instructor.

Computing

The programming assignments will all be done in Python using the NumPy scientific computing library, but prior knowledge of Python is not required. Basic Python will be taught in a tutorial.

You have several options for how to use Python:

- You can install Python yourself on your own machine. For most of you, this will be the most convenient option. (Our assignments will not require especially heavy computation.)
  - Anaconda (`https://store.continuum.io/cshop/anaconda/`) provides a single-click installer for most common platforms, and this is likely the easiest way to install Python and the required libraries.

  - You can install Python (`https://www.python.org/`), NumPy (`http://www.numpy.org/`), and Matplotlib (`http://matplotlib.org/`) manually. This takes a bit more work than using Anaconda.

- You can run it on the CDF machines. All required libraries are already installed. Accounts should have already been created for registered students by the start of the course. If you are having a problem with a CDF account, ask us.

Online forum

We'll use the forum that's part of the Coursera website for this course.

Auditing

If you are not registered in the class, it is possible for you to audit it (sit in on the lectures). Here are the official university rules on auditors (taken from the Department of Computer Science instructor's advice page):

> To audit a course is to sit and listen to the lectures, and perhaps to the tutorials, without formally enrolling. Auditing is acceptable *if the auditor is a student at U of T, and no University resources are to be committed to the auditor.* The "must be a student" condition means that students of other universities, employees of outside organizations (or even of U of T itself!), or any other non-students, are not permitted to be auditors. (If we did not have this rule, the University would require us to collect auditing fees, and we are not willing to do that.)
>
> The "no resources used" condition means that auditors do not get computing accounts, cannot have term work marked, and cannot write exams. In other words, they cannot use instructors time, TA time, or administrative resources of any kind.
>
> An auditor may not attend class *unless there is an empty seat after the last regularly-enrolled student has sat down.* That sounds frivolous, but in fact it is an aspect of an important point: if enrollment in a course has been closed because the room size has been reached, then there may well be physical seats for auditors, because it is rare for every student to appear for a lecture, but auditors will not be allowed to enroll later on in the course, even if some students drop it. Neither instructors nor the department can waive this rule.
>
> Often these conditions are perfectly acceptable to auditors; we don't mean to ban the practice, but only to live within the University's rules.