CSC321 Winter 2017 — Course information
Introduction to Neural Networks and Machine Learning

Course web site: `http://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/`

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. They're 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.

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 final exam. Since both sections are at full enrollment, **please attend your assigned section**.

- **Afternoon section**
  Lectures: Tuesdays and Thursdays, 1:10–2:00pm, in Sidney-Smith, room 1073.
  The first lecture is on January 5; the last lecture is on April 4. There are no lectures on February 21 and 23 (reading week).
  Tutorials: Thursdays 2:10–3:00pm in Bahen 1200. There is no tutorial on January 5.

- **Night section**
  Lectures: Tuesdays, 6:10–7:50 pm, in Bahen, room 1200.
  The first lecture is on January 10; the last lecture is on April 4. There is no lecture on February 21 (reading week).
  Tutorials: Tuesdays, 8:15–9:00pm, also in Bahen 1200. This leaves 25 minutes for a quick dinner between lecture and tutorial. There is no tutorial on April 4.

Detailed topics for each lecture and tutorial can be found on the course web page.

Teaching Team

Instructor
Roger Grosse
e-mail: `csc321prof@cs.toronto.edu`
Office hours: Mon 10am–noon in Pratt 290F

Teaching Assistants:
Yuhuai (Tony) Wu
Renjie Liao
Lisa Zhang
Paul Vicol
Ladislav Rampasek
Chun-Hao Chang

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

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

Prerequisites

Because we make heavy use of calculus, probability, and linear algebra, this course has some substantial prerequisites. All students who did not meet the prerequisites were already removed from the course before the start of the term, so if you haven't heard anything from the Undergraduate Office, that means you've met the prerequisites.

- **Calculus:** (MAT136H1 with a minimum mark of 77)/(MAT137Y1 with a minimum mark of 73)/(MAT157Y1 with a minimum mark of 67)/MAT235Y1/MAT237Y1/MAT257Y1
- **Linear Algebra:** MAT221H1/MAT223H1/MAT240H1
- **Probability:** STA247H1/STA255H1/STA257H1
- **Multivariable calculus (recommended):** MAT235Y1/MAT237Y1/MAT257Y1
- **Programming experience (recommended)**

Load

There are 23 hours of lectures and 10 hours of tutorials.

Readings

There is no required textbook for the class. 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.

- Video lectures for UofT Professor Geoffrey Hinton's Coursera course. Professor Hinton is one of the fathers of the field, so think of these as the Feynman Lectures of neural nets.
  `https://www.youtube.com/playlist?list=PLoRl3Ht4JOcdU872GhiYWf6jwrk_SNhz9`

- *Deep Learning*, a textbook by Yoshua Bengio, Ian Goodfellow, and Aaron Courville.
  `http://www.deeplearningbook.org/`

- Andrej Karpathy's lecture notes on convolutional networks. These are very readable and cover the material in roughly the first half of the course.
  `http://cs231n.github.io/`

- Richard Socher's lecture notes, focusing on RNNs.
  `http://cs224d.stanford.edu/syllabus.html`

- *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`

- Video lectures for Hugo Larochelle's neural networks course. These are similar to Professor Hinton'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.
- Weekly homeworks: 10%.

Academic Integrity

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.

Weekly Homeworks

In order to give you additional practice with the material, we assign weekly homeworks, which give you additional practice with the course content and encourage you to keep on top of the material. Each one consists of 2-3 conceptual questions and is meant to take a few hours.

Dates. Weekly homeworks are due at 11:59pm each Monday, beginning January 16. Each homework covers material up through the Tuesday before it is due.

Format. Weekly homeworks must be submitted in PDF format through MarkUs. These can be typed, but scans of handwritten solutions are also acceptable.

Lateness. Weekly homeworks will be accepted up to 4 days late (i.e. until 11:59pm on Friday), but 15% will be deducted for each day late, rounded up to the nearest day. Any exceptions require an official Student Medical Certificate.

Weighting. In aggregate, the homeworks count for 10% of the total mark for the course, so individually they count for roughly 1% each.

Collaboration policy. You are expected to work on the homeworks 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.

Programming 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 brief report (roughly one page plus figures) that describes the results you obtained.

Dates. See the calendar page for assignment due dates. Typically, they will be due at 11:59pm on Thursdays. Each assignment uses lecture material up through the Tuesday the week before it is due (i.e. 9 days in advance of the deadline).

Format. All programming assignment reports must be handed in as PDFs through MarkUs. They must be formatted using LaTeX.

Lateness. Programming assignments will be accepted up to 4 days late, but 15% will be deducted for each day late, rounded up to the nearest day. Any exceptions require an official Student Medical Certificate.

Weighting. Each of the programming assignments will count for 10% of the total mark for the course.

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, 2/28**. It is closed book. It covers all material for the first 12 lectures, *i.e.* up through the material covered on Tuesday, 2/14. We will link to practice midterms.

1. The midterm for the afternoon section will be held during lecture time, **1:10-2pm on Tuesday, 2/28**. There will be a lecture as usual from 1:10–2pm on Thursday 3/2, but no tutorial this week.

2. The midterm for the night section will be held during lecture time, **6:10-7pm on Tuesday, 2/28**. Afterwards, there will be a 30 minute dinner break, followed by a lecture from 7:30–8:30pm. There will be no tutorial on 2/28.

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 framework, 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 Discourse for the course forum. The URL will be given out on the course web page.

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.