Course Information

CSC413H1S: Neural Networks and Deep Learning (Winter 2024)

Department of Computer Science, University of Toronto

Website: amfarahmand.github.io/NN-Winter2024

Contact: csc413-2024-01@cs.toronto.edu (Do not contact the instructors or TAs individually.)

Forum (Piazza): piazza.com/class/lqyinbo4wzp2n4

Submissions: markus.teach.cs.toronto.edu/2024-01

Lecture	Time	Location	Instructor
LEC 0101/2101	R1-4	SF	Amir-massoud
LEC 0201/2001	T1-4	MC	Amanjit
LEC 5101/2501	T6-9	BA	Rupert

Tutorials are in the third hour of each lecture. See **ACORN** for exact rooms.

Instructors

- Amir-massoud Farahmand
  - Office Hour: TBA
- Amanjit Singh Kainth
  - Office Hour: TBA
- Robert (Rupert) Wu
  - Office Hour: T4-5 @ BA 2272

Lead TA: Claas Voelcker

TAs: TBA

Overview

Machine learning (ML) is a set of techniques that allow computers to learn from data and experience, rather than requiring humans to specify the desired behaviour manually. This course focuses on Neural Networks (NN) models and the Deep Learning (DL) approach to design ML systems. NNs and DL have become increasingly central both in AI as an academic field, and in industry. This course gives an overview of both the foundational ideas and the recent advances in neural network algorithms.

By the end of this course, the students will learn about

- Neural Network Architectures such as Feedforward NN, Convolutional NN, Recurrent NN, and Transformers
- Training and Optimization: Backpropagation, SGD, etc.
- Sequence and language modelling
- Transfer Learning
- Adversarial Attacks
- Generative models such as Generative Adversarial Networks

The students are expected to learn the intuition behind many NN architectures and DL algorithms. Through homework assignments, they will also learn how to implement these methods and use them to solve machine learning problems.

Prerequisites

This is a second course in machine learning, so it has some substantial prerequisites, which will be enforced.

- **Linear Algebra:** MAT223H1/MAT240H1/MAT185H1/MAT188H1/MAT223H5/MATA23H3
- **Multivariate Calculus:** MAT235Y1/MAT237Y1/MAT257Y1/MAT257Y5/MAT291H1/MAT294H1 /AER210H1/(MAT232H5, MAT236H5)/(MAT233H5, MAT236H5)/(MATB41H3, MATB42H3)
- **Machine Learning:** CSC311H1/CSC311H5/CSCC11H3/CSC411H1/STA314H1/ECE421H1/ROB313H1

Lecture Schedule

Week	Date (Monday of)	Lecture Topic
01	Jan 08	Introduction and Review of Linear Models
02	Jan 15	Multi-layer Feedforward NN and Backpropagation
03	Jan 22	Automatic differentiation, distributed representation, and GloVe
04	Jan 29	Convolutional Neural Networks
05	Feb 05	Optimization
06	Feb 12	More on Optimization & Generalization
–	Feb 19	(Reading Week)
07	Feb 26	CNN Features Visualization, Transfer Learning, Adversarial Examples
08	Mar 04	Recurrent Neural Networks
09	Mar 11	Text Generation with RNN and Attention Mechanism
10	Mar 18	Generative Models
11	Mar 25	Generative Adversarial Learning and Fairness/Ethics
12	Apr 01	Special Topic (TBD)

Note on slides: Many have contributed to the design of these slides. Credit goes to several members of the U of T, and beyond, including (recent past, as far as we know): Florian Shkurti, Igor Gilitschenski, Lisa Zhang, Jimmy Ba, Bo Wang, Roger Grosse, and us.

Announcements (Quercus)

The University's offical learning management system (LMS) is Quercus. Our use of this platform will be limited to announcements, which also send notification emails by default. `https://q.utoronto.ca/`

Online Forum (Piazza)

We will use the Piazza course forum as the primary source of interactive and crowd-sourced learning. The expectation is that students search through the forum for similar queries before opening a new one or emailing us. `piazza.com/class/lqyinbo4wzp2n4/`

Accessibility

The CSC413 teaching staff is fully committed to ensuring accessibility for all our students. For the students looking for additional academic accommodations or accessibility services registration, please visit `www.accessibility.utoronto.ca`. Students are encouraged to review the course syllabus (this document) at the beginning of a course and discuss questions regarding their accommodations for the course with their Accessibility Advisor. Once registered, students should send the Letter of Academic Accommodations to our ticketing system at `csc413-2024-01@cs.toronto.edu` as soon as possible by Friday, January 26th, 2023.

Course Enrollment

CSC413/2516 always had long waiting lists for the last few years. The hard enrollment cap is determined by teaching resources available at the department level. Note that waitlists typically expire one week after the course starts. Once waitlists are removed, students are responsible for trying to enroll in the course on ACORN in a first-come, first-serve fashion. If you have further questions, please get in touch with CS undergrad office or CS graduate office.

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.

Deliverables and Policies

Item	Weight
Mathematical Homeworks * (×2)	8% (4% each)
Programming Homeworks * (×8)	32% (4% each)
Readings (×5)	10% (2% each)
Project Proposal **	10%
Project Report and Code **	20%
Take-Home Test	20%

* Individual or in groups of two (2).

** In groups of three or four (3-4).

Due Dates Schedule

This is a **tentative** schedule of the homework assignments, and the exact dates are subject to change according to the progress of the course. You have at least one week after the release date to submit your solutions.

Tutorial/Assignment	**Post Date**	**Due Date**
Lab 1: Linear Models	Jan 16	**Jan 26**
Math 1: Backpropagation	Jan 16	**Jan 26**
Lab 2: Multi-Layer Perceptrons with MedMNIST	Jan 23	**Feb 2**
Lab 3: Word Embeddings	Jan 30	**Feb 9**
Project Proposal	Feb 02	**Feb 16**
Lab 4: Differential Privacy	Feb 13	**Mar 1**
Math 2:	Feb 27	**Mar 8**
Lab 5: Transfer Learning and Descent	Mar 05	**Mar 15**
Lab 6: GradCAM and Input Gradients	Mar 12	**Mar 22**
Paper Reading Assignments (first two)	Jan 19	**Mar 15**
Lab 7: Text Classification using RNNs	Mar 19	**Mar 30**
Paper Reading Assignments (last three)	Jan 19	**Apr 2**
Take-Home Test	Apr 2	**Apr 4**
Lab 8: Text Generation with Transformers	Mar 26	**Apr 5**
Project Report & Code	Mar 26	**Apr 19**

- All HW assignments (Lab or Math) are due on the Friday of the week after they are released.
- Unless otherwise indicated, due times are at 5PM (Toronto time).
- The teaching team reserves the right to release homeworks early without prior notice.
- Should any handout (except for the test) be posted late, the due date will be at least 7 days after the post time.
- All deliverables should be submitted via MarkUs: `markus.teach.cs.toronto.edu/2024-01/`.

Late Policies

Late Submissions (assignments, proposals, reports, etc)

- Submissions should be handed in by deadline; a late penalty of 10% per day will be assessed thereafter (up to 3 days, then submission is blocked).
- Extensions will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the course coordinator at least one week before the due date.

Homeworks

There are ten homeworks worth 4% each: two (2) mathematical and eight (8) programming. These are to be completed individually or in groups of (2) students subject to the collaboration policy.

Research Project

The final project component will take the form of a research paper. Students will propose (10%) and conduct an investigation and/or devise a method (20%) that leads to a report and codebase that could reasonably be submitted to an academic journal, conference or workshop. Details about format and length will be shared at a later date. Students are expected to work in a group of three or four (3-4) subject to the collaboration policy.

Reading Assignments

The following papers (to be posted) are a combination of seminal papers in DL, topics that we didn't cover in lectures, or active research areas. You need to choose **five (5)** papers out of them, depending on your interest. We will post the papers as the course progresses (so check here often). Please read them and try to understand them as much as possible. It is not important that you completely understand a paper or go into detail of the proofs (if there is any), but you should put some effort into it.

After reading each paper:

- You should summarize it in a short paragraph (100-200 words). Highlight the main points of the paper. Ignore the less interesting aspects.
- Try to come up with one or two suggestions on how the method/idea described in the paper can be used or extended.

Note: This list only covers a tiny fraction of all great papers published in various ML/DL venues. Many of our favourites are not included.

There are two deadlines for Reading Assignments:

- Mar 15, for which you submit your summary of two papers.
- Apr 2, for which you submit the summary of three other papers.

Tests

There will be a take-home test worth 20% of the course grade to to be held on Thursday, April 4th. It covers material up through Lecture 11 (one week prior to the test). The midterm test will be accessible on MarkUs for a 24-hour duration.

Missing the Test

A missed test without excuse will receive a grade of 0. In the event of illness, students should contact us at `csc413-2024-01@cs.toronto.edu` before the test date and approved by the instructors. We will arrange a make-up test within 7 days of the original midterm time.

Academic Integrity

By the time you get to an advanced course like CSC413 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 except for the final project. This means, **any work or ideas that are not your own must be acknowledged with a citation**. The University of Toronto's Code of Behaviour on Academic Matters outlines the behaviours that constitute academic misconduct, the processes for addressing academic offenses, and the penalties that may be imposed. You are expected to be familiar with the contents of this document.

Collaboration

Collaboration on the Homework Assignments **is allowed**, under certain conditions:

- You can discuss the assignment with another student (group of two).
- In your submission, you need to be very clear about the contribution of each individual. For example, you should say we did a pair-programming or person A solved this part while person B solved another part.
- You *can* use copilot, ChatGPT, etc. to solve the problem, but that would consider as your group member. That is, you can have either a 2-human group or 1 human + 1 machine group (no 2 machine group). If you do it, you should report it as well.

You need to form a team of 3-4 members to work on your projects (the exact number will be determined after finalizing the number of students enrolled).

- Similarly to the homework assignments, you need to report the contribution of each collaborator.
- If you get the help of a machine, you need to clearly indicate that. The machine will likewise cost you one of the team members.

Collaboration on the Take-home Test or Paper Readings is **not allowed**. These should be done as individual.

Computation Resources

Many of the deep learning success stories in the recent years rely on the advances of modern GPU computing. The programming assignments here are lightweight comparing to the state of the art deep learning models in terms of their computation requirement. But we highly recommend you to debug your models and to complete the experiments on a modern GPU. Here are the list of free resources you have access to:

- **Google Colaboratory**: A web-based iPython Notebook service that has access to a free Nvidia T4 GPU per Google account. PyTorch (recommended) and TensorFlow are both natively supported. Enhanced resources available through Kaggle. Recommended for homeworks and some projects.
- **Department Teaching Labs**: Linux compute servers with desktop or datacentre-class GPUs. Recommended for projects.
- Google Compute Engine: GCE delivers virtual machines running in Google's data center. Recommended for course project.

Software

For the homework assignments, we will use Python, and libraries such as NumPy, SciPy, and scikit-learn. You have two options:

- All the required packages are already installed on the Teaching Labs machines and Colab.
- If you want to set up your own environment, we recommend that you use `virtualenv` or `conda` to create an environment and install the required packages. For example:

```
# if you are using conda
conda create --name csc413
source activate csc413

# if you are using virtualenv
virtualenv csc413
source csc413/bin/activate

# in both cases, install packages this way
pip install scipy numpy autograd matplotlib jupyter sklearn
```

Useful Resources

This course does not have a required textbook, but the following resources can be useful:

- Yoshua Bengio, Ian Goodfellow, and Aaron Courville, Deep Learning
- Dive into Deep Learning
- Geoffrey Hinton's course.
- Andrej Karpathy's lecture notes on convolutional networks.
- Richard Socher's lecture notes, focusing on RNNs.
- Hugo Larochelle's neural networks course.

Other Useful Resources for Machine Learning:

If you need to brush up your basic knowledge of ML, you can take a look at one of the previous offerings of it at the U of T. For example, CSC2515 - Fall 2022 (the content is almost the same as CSC311).

- Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning, Second Edition, 2009.
- Christopher M. Bishop, Pattern Recognition and Machine Learning, 2006
- Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, Second Edition, 2018.
- Amir-massoud Farahmand, Lecture Notes on Reinforcement Learning, 2021.
- Kevin Murphy, Machine Learning: A Probabilistic Perspective, 2012.
- Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, An Introduction to Statistical Learning, 2017.
- Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, 2014.
- David MacKay, Information Theory, Inference, and Learning Algorithms, 2003.