



CSC411: Machine Learning and Data Mining (Winter 2017)

About CSC411

This course serves as a broad introduction to machine learning and data mining. We will cover the fundamentals of supervised and unsupervised learning. We will focus on neural networks, policy gradient methods in reinforcement learning. We use the Python NumPy/SciPy stack. Students should be comfortable with calculus, probability, and linear algebra.

Required math background

Here's (roughly) the math that you need for this course. Linear Algebra: vectors: the dot product, vector norm, vector addition; matrices: matrix multiplication. Calculus: derivatives, derivatives as the slope of the function; integrals. Probability: random variables, expectation, independence. Other topics will be needed, but are not part of the pre-requisites, so I will devote an appropriate amount of lecture time to them.



Nikulin & Novak, [Exploring the Neural Algorithm of Artistic Style](#)

Evaluation

The better of:

40%: Projects
30%: Midterm
30%: Final Exam

Or:

40%: Projects
20%: Midterm
40%: Final Exam

You must receive at least 40% on the final exam to pass the course.

Teaching team

Instructor: Michael Guerzhoy, **Office:** BA5244, **Email:** [guerzhoy at cs.toronto.edu](mailto:guerzhoy@cs.toronto.edu) (please include CSC411 in the subject, and please ask questions on [Piazza](#) if they are relevant to everyone.)

CSC411 TAs: Patricia Thaine, Aditya Bharagava, Yulia Rubanova, Yingzhou Wu, Shun Liao, Bin Yang, Jixuan Wang, Aleksei Shkurin, Alexey Strokach, Katherine Ge, Karo Castro-Wunsch.

Projects

A sample report/LaTeX template containing advice on how to write project reports AI courses is [here](#) (see the [pdf](#) and [tex](#) files). (The example is based on Programming Computer Vision, pp27-30.) Key points: your code should generate all the figures used in the report; describe and analyze the inputs and the outputs; add your interpretation where feasible.

[Project 1: Face Recognition and Gender Classification Using Regression \(10%\)](#) Due ~~Jan. 29~~ Feb 1 at 11PM

[Project 2 \(10%\)](#) Due Feb. 19 at 11PM

[Project 3 \(10%\)](#) Due Mar. 19 at 11PM

[Project 4 \(10%\)](#) Due Apr. 2 at 11PM

Lateness penalty: 5% of the possible marks per day, rounded up. Projects are only accepted up to 72 hours (3 days) after the deadline.

Midterm

Friday March 3, 6pm-8pm. Make-up midterm for those who have a **documented** (a screenshot and/or explanatory email is sufficient) conflict with the main timeslot: TBA. Please send me email if there is an issue.

Coverage: the lectures and the projects, focusing on the lectures.

Getting help

Michael's office hours: Monday 6-7PM, Wednesday 5-6PM in BA5244. Or email for an appointment. Or drop by to see if I'm in. Feel free to chat with me after lecture.

[Course forum on Piazza](#)

Piazza is a third-party discussion forum with many features that are designed specifically for use with university courses. We encourage you to post questions (and answers!) on Piazza, and read what other questions your classmates have posted. However, since Piazza is run by company separate from the university, we also encourage you to read the privacy policy carefully and

only sign up if you are comfortable with it. If you are not comfortable with signing up for Piazza, please contact me by email to discuss alternative arrangements.

Study Guide

[The CSC411 study guide \(continuously updated\)](#)

Software

We will be using the Python 2 NumPy/SciPy stack in this course. It is installed in the Teaching Labs.

For the first two projects, the most convenient Python distribution to use is [Anaconda](#). If you are using an IDE and download Anaconda, be sure to have your IDE use the Anaconda Python.

I recommend the Pyzo IDE available [here](#). To run Pyzo in the Teaching Labs, simply type `iep` in the command line.

We will be using Google's [TensorFlow](#) later on in the course.

We will be using the CS Teaching Labs and AWS for GPU computing. Please sign up using [AWS Educate](#). Use your CS Teaching Labs account when signing up for AWS Educate. You can use either the GPU-equipped machines in the Teaching Labs (30 machines in BA2210 and 6 machines in BA3200) or AWS.

Resources

[Pattern Recognition and Machine Learning](#) by Christopher M. Bishop is a very detailed and thorough book on the foundations of machine learning. A good textbook to buy to have as a reference.

[The Elements of Statistical Learning](#) by Trevor Hastie, Robert Tibshirani, and Jerome Friedman is also an excellent reference book, available on the web for free at the link.

[An Introduction to Statistical Learning with Applications in R](#) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani is a more accessible version of The Elements of Statistical Learning.

[Deep Learning](#) by Yoshua Bengio, Ian Goodfellow, and Aaron Courville is an advanced textbook with good coverage of deep learning and a brief introduction to machine learning.

[Learning Deep Architectures for AI](#) by Yoshua Bengio is in some ways better than the Deep Learning book, in my opinion.

[Reinforcement Learning: An Introduction](#) will be useful when we discuss reinforcement learning

[Geoffrey Hinton's](#) Coursera course contains great explanations for the intuition behind neural networks.

[The CS229 Lecture Notes](#) by Andrew Ng are a concise introduction to machine learning.

[Andrew Ng's Coursera course](#) contains excellent explanations.

[CS231n: Convolutional Neural Networks for Visual Recognition](#) at Stanford ([archived 2015 version](#)) is an amazing advanced course, taught by Fei-Fei Li and Andrej Karpathy (a UofT alum). The course website contains a wealth of materials.

[CS224d: Deep Learning for Natural Language Processing](#) at Stanford, taught by Richard Socher. CS231, but for NLP rather than vision. More details on RNNs are given here.

[Python Scientific Lecture Notes](#) by Valentin Haenel, Emmanuelle Gouillart, and Gaël Varoquaux (eds) contains material on NumPy and working with image data in SciPy. (Free on the web.)

Announcements

All announcement will be on Piazza.

Lecture notes

Coming up: introduction to Numpy/SciPy, k-Nearest Neighbours, linear regression and gradient descent. See Andrew Ng's coursera course Weeks 1 and 2, [Notes, part 1](#) from CS229, and Hastie et al. 2.1-2.3. Gradients: [Understanding the Gradient](#), [Understanding Pythagorean Distance and the Gradient](#)

The needed multivariate calculus: see the first three videos [here](#).

Week 1: [Welcome to CSC411, K Nearest Neighbours, Linear Regression](#).

NumPy demos: [NumPy demo](#), [NumPy and images \(bieber.jpg\)](#), [plotting functions of 2 variables in NumPy](#).

[Gradient Descent for functions of one variable](#).

Q: for functions of two variables, what's the unit vector that points in the "downhill" direction on the surface plot? (answer to be posted later.)

Coming up: Gradient descent in multiple variables, linear classifiers, multiple linear regression (see Weeks 1-3 from Andrew Ng's course). Learning with maximum likelihood.

Week 2: [2D gradient descent](#), [Gradient descent for linear regression \(galaxy.data\)](#)

[Linear classification](#), [Linear regression 2](#), maximum likelihood (on the board)

Coming up: Inference using maximum likelihood and Bayesian inference (see [here](#)). Overfitting in regression. Intro to Neural Networks (Bishop Ch. 5, Week 4 from Andrew Ng's course.)

Week 3: [Overfitting and linear regression](#), [Intro to Bayesian inference](#).

[maximum likelihood](#), [bayesian inference](#), [neural networks intro](#).

Just for fun: The first joke on Yann LeCun's Geoff Hinton Facts page is [relevant](#).

Coming up: More neural networks, backpropagation. See the Bishop chapter on Neural Networks, week 5 of Andrew Ng's course.

Video tutorial: [Forward propagation: setup](#), [Forward propagation: Vectorization](#), [Backprop for a specific weight, part 1](#), [Backprop for a specific weight, part 2](#), [Backprop for a generic weight](#), [Backprop for a generic weight: vectorization](#)

Week 4: Multilayer neural networks: [One-hot encoding](#), [Gradients of neural networks and Backpropagation](#), [activation functions](#).

Training neural networks: [Vectorizing neural networks](#), [Overfitting; preventing overfitting](#).

Interpreting neural networks: [How neural networks see](#).

Coming up: Overfitting and preventing overfitting, continued. Optimizing neural networks. Convolutional neural networks.

Demo: [TensorFlow playground](#).

Week 5: Overfitting: [Overfitting](#); [preventing overfitting](#). [Optimizing neural networks](#), [intro to convolutional networks](#), [modern ConvNet architectures](#).

Just for fun: [Neural correlates of interspecies perspective taking in the post-mortem Atlantic Salmon](#)

MarkUs

All project submission will be done electronically, using the *MarkUs* system. [Log in to MarkUs](#).

For group projects, to submit as a group, one of you needs to "invite" the other to be partners, and then the other student needs to accept the invitation. To invite a partner, navigate to the appropriate Assignment page, find "Group Information", and click on "Invite". You will be prompted for the other student's CDF user name; enter it. To accept an invitation, find "Group Information" on the Assignment page, find the invitation listed there, and click on "Join". **Only one student must invite the other:** if both students send an invitation, then neither of you will be able to accept the other's invitation. So make sure to agree beforehand on who will send the invitation! Also, remember that, when working in a group, only one person must submit solutions.

To submit your work, again navigate to the appropriate Exercise or Assignment page, then click on the "Submissions" tab near the top. Click "Add a New File" and either type a file name or use the "Browse" button to choose one. Then click "Submit". You can submit a new version of any file at any time (though the lateness penalty applies if you submit after the deadline) — look in the "Replace" column. For the purposes of determining the lateness penalty, the submission time is considered to be the time of your latest submission.

Once you have submitted, click on the file's name to check that you have submitted the correct version.

LaTeX

Web-based LaTeX interfaces: [Overleaf](#), [ShareLaTeX](#)

[TeXworks](#), a cross-platform LaTeX front-end. To use it, install [MikTeX](#) on Windows and [MacTeX](#) on Mac

[Detexify²](#) - LaTeX symbol classifier

The [LaTeX Wikibook](#).

Additional [LaTeX Documentation](#), from the home page of [the LaTeX Project](#).

[LaTeX on Wikipedia](#).

Policy on special consideration

Special consideration will be given in cases of documented and serious medical and personal circumstances.

