Course Administration

Prof. Phil Schniter

The Ohio State University

ECE 4300: Introduction to Machine Learning, Sp20

Outline

- Syllabus summary
- Software tools

Instructors

- Prof. Phil Schniter (schniter.1@osu.edu)
  - Office: Dreese Labs 616
  - Office hrs: TBD
- TA: Ms. PG Guan (guan.219@buckeyemail.osu.edu)
  - Office hrs: TBD
  - In charge of grading homeworks and labs

Course learning objectives

- Understand what machine learning is
  - Regression, classification, clustering, dimensionality reduction
- Understand the *mathematics* behind standard models, algorithms, and methods
  - Including concepts from linear algebra, probability, statistics.
- Gain experience using standard software tools:
  - Python, Numpy, matplotlib, sklearn
  - PyTorch
  - GitHub, git

Prerequisites

- Undergrad status in ECE major
- Math 2568 (calculus and linear algebra)
  - Vectors, matrices, partial derivatives, gradients
  - We will review as needed
- Stats 3470 (probability and statistics)
  - Basic concepts such as mean, variance, correlation, probability densities, conditional distributions, Gaussian distribution
  - We will review as needed
- CSE 1222 or ENGR 1281 (basic programming)
  - Basic programmig such as C/C++ and Matlab
  - No Python background is assumed

Optional textbooks

- G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 2013.
  - Free at `http://faculty.marshall.usc.edu/gareth-james/ISL/`
- S. Raschka, *Python Machine Learning, 3rd Ed.*, 2019.
  - $5 at https://www.packtpub.com/
- A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, 2019.
- J. VanderPlas, *Python Data Science Handbook*, 2017.
  - Free at `https://jakevdp.github.io/PythonDataScienceHandbook/`
- E. Stevens and L. Antiga, *Deep Learning with PyTorch*, 2020.
  - Free at `https://www.manning.com/books/deep-learning-with-pytorch`

Grading

- Components:
  - Midterm 1: 20%
  - Midterm 2: 20%
  - Weekly homeworks & labs: 40%
  - Final Project: 20%
- Homeworks: traditional analytical assignments
- Labs: python jupyter notebooks
- Midterm exams: closed-book with cheatsheet
- Final project:
  - Apply concepts from this class in an independent way
  - Must use data & Python, and provide written report
  - Teams of 3 people preferred

Outline

- Syllabus summary

- Software tools

GitHub

- Course materials hosted at `https://github.com/ece4300sp20/ece4300`
- This is a private repository; you'll need to get access
  - See detailed instructions in Carmen announcement
- Can view most materials with a web browser. But to run demos, you'll need to clone the repository to your local machine (and keep it updated)
- To clone/update, you'll need to run a git client on local machine
  - See detailed instructions in course repo
  - Also, many tutorials on web
- Will also use GitHub/git to submit labs

Python

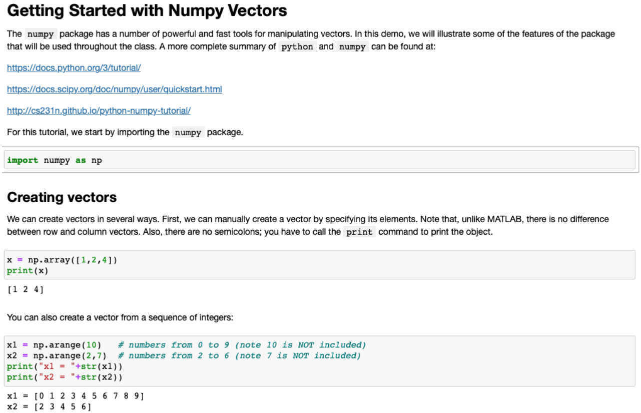
- Python
  - A general and powerful language
  - Contains many machine-learning libraries, especially for deep learning
  - Free and open source!
- Prerequisites
  - No Python background is assumed; we'll teach it in this course
  - Some programming experience required, though (e.g., C/C++, Matlab)
  - Experience with objected-oriented programming (e.g., C++) is useful
- The course repo includes primers on . . .
  - Basic use of Python and Numpy
  - Moving from Matlab to Python (if you are experienced with Matlab)

Python environments

- The labs in this course will use Python and relevant libraries
  - Python 3.x (not 2.x!)
  - Numpy, sklearn
  - Juptyer notebook
  - PyTorch (later in the course)
- You can run Python on your personal machine or in the computer labs
  - I recommend installing the Anaconda distribution
- You can also run it in the cloud
  - Amazon web services (AWS) `https://aws.amazon.com`
  - Google cloud platform (GCP) `http://cloud.google.com`
  - Microsoft azure https://azure.microsoft.com

Jupyter notebook demos

- The lecture slides will make use of Jupyter notebook demos.
- The github repo includes a demo on getting started with Numpy:



Getting Started with Numpy Vectors

The `numpy` package has a number of powerful and fast tools for manipulating vectors. In this demo, we will illustrate some of the features of the package that will be used throughout the class. A more complete summary of `python` and `numpy` can be found at:

https://docs.python.org/3/tutorial/

https://docs.scipy.org/doc/numpy/user/quickstart.html

http://cs231n.github.io/python-numpy-tutorial/

For this tutorial, we start by importing the `numpy` package.

```
import numpy as np
```

Creating vectors

We can create vectors in several ways. First, we can manually create a vector by specifying its elements. Note that, unlike MATLAB, there is no difference between row and column vectors. Also, there are no semicolons; you have to call the `print` command to print the object.

```
x = np.array([1,2,4])
print(x)
```

```
[1 2 4]
```

You can also create a vector from a sequence of integers:

```
x1 = np.arange(10)   # numbers from 0 to 9 (note 10 is NOT included)
x2 = np.arange(2,7)  # numbers from 2 to 6 (note 7 is NOT included)
print("x1 = "+str(x1))
print("x2 = "+str(x2))
```

```
x1 = [0 1 2 3 4 5 6 7 8 9]
x2 = [2 3 4 5 6]
```