0 - Course Introduction

November 6, 2020

1 Programming for Data Science and Artificial Intelligence

1.1 0 Course Introduction

- My name is Chaklam. Born Hong Kong. Research Interests on Neurosciene and Machine Learning. Welcome you all! This course will be fun and super tough!
- All materials can be found on my **Github - https://github.com/chaklam-silpasuwanchai/**. If you forget, you can also get the link from http://chaklam.com
- I am sitting in either room **101** or **203** most of the time. Please send emails to my calendar - chaklam@ait.asia for appointment
- Room 203 also has many machine learning books I used for this class. You are not allowed to take them away - but you are certainly welcomed to read them at my lab mini library.
- For online folks, you can always schedule a zoom meeting with me by sending me emails.
- Introducing our TA, Mr. Akrardet (Jo) - st121413@ait.asia PhD. students in my lab working on brain computer interfaces. He will be in charge of
  - Google classroom (code: will be provided by Jo)
  - Checking assignments
  - Uploading our lecture video to some repository
  - Organizing quiz
  - Troubleshoot any of your problems during class (but please be gentle to him!)

1.1.1 Setting up your Python and Jupyter

- I use **python3** as interpreter.
- We will be using a lot of modules that may not be pre-built in python, so I assume you know how to use pip to install missing modules. I also assume you use linux systems since I have no experience in Windows :) Read Lab 0 if you haven't.
- Highly recommend to use **venv** to setup your python environment, before installing python modules. If you do not know how and what is venv, read this. https://docs.python.org/3/library/venv. Docker is welcomed as well.
- Sometimes, it will take like several hours (like 3 to 4 hours) to run your model. CSIM has some resources which you can use to help your model train faster. One is the puffer system. You can access it using https://puffer.cs.ait.ac.th (in CSIM), which got four 2080Tis, 64GBs Ram, and very powerful CPU (I don't remember the series). You must use your CSIM account to

access and has been allocated a certain space amount. Note that ~jovyan/ is all temporary, going away when the user's server is shut down. ~jovyan/work is persistent. All notebooks are automatically stored in /work so do not worry your file will be lost. The current server is a work-in-progress so make sure you backup! There are also some PCs you can use at CSIM lab which has one NVIDIA 2060 super, i7, so that's plenty for you to run your model as well.

1.1.2 Pre-requisites / Expectations

- This is **NOT a sklearn/pyTorch plug-and-play course**. I don't teach **applied stuff**...it is NOT intended to teach you to quickly get some dirty models running (you may want to try Rapidminer or PowerBI instead which is probably faster than Python; also I would recommend many nice Youtube videos out there). Instead, this course aims for **technical** audience who would like to develop fundamental skills needed to link math, data sicence, and python seemlessly together. There are other courses such as **Business Intelligence** that aims more toward **business** audience and focus more on using such tools.
- This is definitely not one study-and-become-expert course, but rather **prepare you to able to read research papers and translate them into Python with ease**. Thus, everything in our course will be **coding from scratch** and **mathemtical-based**. Of course, I will also have a self-study lecture section where I talked about sklearn/PyTorch but it's intended only for self-study, not anything I will talk about.
- **Math, especially knowledge in linear algebra, differential calculus, and gaussian statistics are required. I assume that you know most of them, but I will try to be gentle in the beginning and give you guys a quick recap whenever I talk about a related math topic for the first time**. For your own deeper study, I recommend the followings:
  - Linear Algebra short recap - https://www.cs.cmu.edu/~zkolter/course/15-884/linalg-review.pdf; Recommended book - http://linear.axler.net; Recommended playlist (Prof. Gilbert Strang) - https://www.youtube.com/watch?v=7UJ4CFRGd-U&list=PLE7DDD91010BC51F8
  - Stats - http://greenteapress.com/thinkstats/ (free!)
- I have prepare some tutorials you must self-study with the beginning "0". Please look through them as it will be helpful to understand the course.
- This course is best taken together with Machine Learning with Prof. Matthew Dailey, where you can study the full derivation of formulas. We shared a lot of similarities such as regression, and classification, but with a stronger focus on the integration between mathematics and python programming.
- Very practical course. **Intermediate level**. Could be **extremely** difficult for those without any programming experience. Requires **huge huge amount of self-study hours**, especially if you are weak in programming/math. I highly recommend to read the materials before you come to class; most of my students found them very helpful. You WON'T BE GOOD if you only listen to my lecture, and don't practice.

1.1.3 What kind of *ouptput* students I want

I have actually high expectations of myself to develop strong mathematical and programming skills in students. Here are what I expect the students who finish my course will look like:

- Able to **code python fluently**, with good object-oriented (OO) and software engineering (SE) principles. In a sense, students should be able to write class in a "clean", "efficient", and "pythonic" way.
- Able to **self-study** advanced python stuffs in the future by increasing literacy in reading APIs, documentations, etc. Thus, I encourage students to read APIs frequently, instead of just copying answers from StackOverflow.
- Able to **read mathematics** on any given research papers without too much difficulty. Thus I shall cover machine learning algorithms from the perspectives of mathematics, as a mean for you to get acquainted with mathematics.
- Able to **translate those mathematical knowledge to python programming**. Thus, our class will focus on coding from scratch, rather than using library such as sklearn or pytorch.
- Able to understand what are some useful machine learning algorithms, how to implement them, and most importantly, *when* or *when NOT* to use them. Thus I shall regularly compare and contrast different algorithms, i.e., weakness and strengths.
- Able to **self-explore, self-learn beyond what I teach, and link all the knowledge** you have, and combined with your own unique background (e.g., economics, medicine, finance, retail) to solve a particular data problem. More importantly, able to *judge whether a problem is worth solving*. Thus I shall hope that project is a good starting point.

1.1.4 Resources

- Reference books:
  - [GERON] Geron, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**, 2019 (2nd edition) (https://github.com/ageron/handson-ml2)
  - [VANDER] VanderPlas, J. **Python Data Science Handbook: Essential Tools for Working with Data**, 2016 (1st edition) (https://jakevdp.github.io/PythonDataScienceHandbook/)
  - [HASTIE] Hastie, T., Tibshirani, R., and Friedman, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**, 2016 (2nd edition) (https://web.stanford.edu/~hastie/Papers/ESLII.pdf_)
  - [WEIDMAN] Weidman, S. **Deep Learning from Scratch: Building with Python from First Principles**, 2019 (1st edition) (https://github.com/SethHWeidman/DLFS_code)
  - [CHARU] Charu, C. **Neural Networks and Deep Learning: A Textbook**, 2018 (1st edition)
  - [Deisenroth] Deisenroth, M.P., Faisal A.A., Ong, C.S., **Mathematics for Machine Learning**, 2020 (1st edition) (https://mml-book.github.io/book/mml-book.pdf)
- Online resources:
  - Python tutorials available online: https://docs.python.org/3/tutorial/
  - Jupyter notebook tutorials available online: https://ipython.org/documentation.html

  - Numpy tutorials available online:https://numpy.org/doc/stable/
  - Pandas tutorials available online: https://pandas.pydata.org/docs/
  - Nltk tutorials available online: https://www.nltk.org
  - Matplotlib tutorials available online: https://matplotlib.org/contents.html
  - Scikit-learn tutorials available online: https://scikit-learn.org/stable/user_guide.html
  - Pytorch tutorials available online: https://pytorch.org/tutorials/
  - Nice visuals on CNN - https://github.com/vdumoulin/conv_arithmetic

1.1.5 Course Outline

- 1 Intro to Python*
- 2 Numpy*
- 3 Pandas*
- 4 Matplotlib*
- 5 Supervised Learning: Regression
  - Linear regression
    - Closed Form
    - Batch Gradient Descent
    - Mini-Batch Gradient Descent
    - Stochastic Gradient Descent
  - Polynomial regression
  - Regularlization
    - Ridge regression
    - Lasso regression
    - Elastic regression
- 6 Supervised Learning: Classification
  - Logistic regression
  - Naive Gaussian
  - Support Vector Machine
  - K-Nearest Neighbors
  - Decision Trees
  - Bagging
  - Random Forests
  - AdaBoost
  - Gradient Boosting
- **Midterm / Project starts**
- 7 Unsupervised Learning: Clustering
  - K-means
  - Gaussian mixture
- 8 Unsupervised Learning: Dimensionality reduction
  - Principal Component Analysis
- 9 Deep Learning
  - Neural Network
  - Deep Neural Network
  - Momentum, Decay Learning Rate, Glorot Initialization
  - Convolutional Neural Network
  - Recurrent Neural Network

  - PyTorch (as materials for self-study)
- **Special Talk by my lab student (AIT Brain Lab): Transfer Learning, Natural Language Processing/Text Mining, Transformers, GANs, Autoencoders, Blind Source Separation**
- **Final Project Presentation (one week before final exam)**
- **Final Exam**

Notes

Note 1: * refers to 4 assignments

Note 2: other topics have self-study exercises and solution you can work on. Found in the assignment folder. Highly recommended.

Note 3: some topics may take more than one week to finish; if it takes more time, I shall have additional lecture days - no skipping any topics!

1.1.6 Grading Percentage

- Midterm: 25
- Final: 35
- Project 20
- Assignment 10
- Quiz 10
- Bonus 5

Note: follows gaussian distribution

Note2: I provide special bonus points for those who can identify bugs in my code, or mathematical errors in my formulas. (I don't give points to typos but let me know as well!) This is the incentive for those who carefully read my lecture notes.

1.1.7 Assignments

- There will be four assignments in the beginning of the four topics - Python, Numpy, Pandas, Matplotlib, to get you acquainted with Python. For the rest of topics (starting from regression onward), there will be self-study exercises found in the assignments folder which I highly encourage you to do. All assignments including self-study exercises have solution, found in the assignments folder as well.
- All homework should be **submited via Google classrooms (code: 2ezh4cj)**. You ARE NOT ALLOWED TO COPY my solution, or your friend solution. Both copier and copee shall be given zero without any questionning.
  - Homework is due after **two weeks** of notice. No late work is accepted.
  - All assignments should be supplemented with comments where appropriate.
  - All assignments must be submitted in .pdf for easy checking
- **Does not require you to code exactly like mine**. There are zillion ways of programming python to achieve the same thing. That's the beauty of python! But of course, I would prefer you writing in a pythonic way (i.e., using list comprehension)

1.1.8 Quiz

- TA will carry 3-5 quizzes in the entire semester in the Google classroom. The quizzes are multiple-choice problems and will be conducted using [10 - 120] minutes in the beginning of the class. Jo has the privilege to make it a surprise quiz or informed quiz, in whatever way he desires to.
- If you miss the quiz, there is no make-up

1.1.9 Midterm and Final

- Mid-term and final will be **open book, open internet**, practical exams
- No late exams are accepted

1.1.10 Project

- Project will be judged according to the **technical challenge**, **completeness (it works!)**, and **paper quality**.
  - **Technical challenge** (33%)
    - The problem you choose must be **challenging**. I am NOT interested in anything that simply copy from some medium.com tutorials or kaggle.com.
    - I expect more than 100 hours on the project itself, including self-exploration, investigation of githubs, reading papers, and crafting the software/hardware
    - You are NOT restricted to only models I have taught you. In fact, I am keen (and somewhat expected) to see how you will be exploring more advanced data science domains: autoencoders, GANs, transfer learning, reinforcement learning
    - Since I am working in Brain related areas, I am biased towards anything related to EEG or neuroscience. You can freely use the EEG headset in my brain lab at room 203. Seniors including Jo are there to help you I am personally interested in
      - Emotion Recognition using GANs, Transfer Learning, Autoencoders
      - Brain Spellers using GANs
      - Artifact Removal using Autoencoders, Blind Source Separation
      - Real-time emotion recognizer, real-time BCI spellers
  - **Completeness** (33%)
    - Completeness depend on the project. If it is application problem, it should at least provie a good proof of concept via demos through software + hardware. If it is a modeling problem, it should have a good pipeline from preprocessing, feature extraction, feature engineering, cross validation, modeling, deploying
  - **Paper quality** (33%)
    - You will be expected to write a final report to be submitted on the same day as the final presentation - soft copy submitted at the Google Classroom.
    - Paper writing in **IEEE format**, which should include Introduction, Related Work (at least 5 papers review), Implementation and Architectures, Evaluation, Discussion and Conclusion Recommend using latex format. Allowed format (.docx or .tex). Format can be downloaded here (https://journals.ieeeauthorcenter.ieee.org/create-your-ieee-journal-article/authoring-tools-and-templates/tools-for-ieee-authors/ieee-article-templates/ - choose journal)
    - You will be judged based on the contribution of your work, how well you have reviewed the academic journals/papers, how well you execute your measurements,

and the overall writing rigour

- Deliverables include
  - IEEE paper
  - Python code. Allowed format (.py or .ipynb)
  - PPT Presentation (ppt or pdf)
    - Final presentation - one week before the final exam
  - All submitted via Google classroom
- Team of **2 - 3 people** (special considerations will be given case-by-case basis)

[ ]: