JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

MACHINE LEARNING II
EN.553.741

Spring 2022

https://www.jhu.edu

Instructor:	James Schmidt	**Time:**	1630–1745 MW	**Office Hours:**
Email:	aschmi40@jhu.edu	**Place:**	Hodson 316	After class Wednesday

Description:

Machine Learning describes a mishmash of computational techniques for "finding patterns in data." The scope of use, analytic tools, algorithms, and results are almost too numerous to meaningfully batch all such applications under a common appellation. Still, we try. This course is the second of a two-semester sequence which focuses on implementation. While 740 emphasizes theory (probability, concentration bounds, guarantees for learnability (PAC), and the relevant functional analsyis), this course emphasizes particular hypothesis classes–such as neural networks, support vector machines, reproducing kernel Hilbert spaces–and algorithms to train them. While performance guarantees do not take priority in this course, they will still appear from time to time, and you should be comfortable with material from 740.

In this course, you will learn how a variety of machine learning algorithms work, how to implement them, and how to use libraries which have already implemented them. Because of both their ubiquity in modern machine learning practice as well as their versatility, much of our attention will center around deep neural networks. You will learn to build a fully connected network from scratch, implement backprop directly, and tie in various other aspects into the machine learning pipeline. After learning how to "do things from the ground up," you will also learn how to implement such using Torch.

Prerequisites:

Working knowledge of—or aptitude and willingness to learn as needed—probability theory, multivariable calculus, analysis, linear algebra, and python, and fundamentals of machine learning as taught in 740

Git Repository:

https://github.com/schmidttgenstein/sp23-mlii.git

Assignments, data, and sometimes starter code will be available on this repository. While we will not be engaging the full power of git with source control and shared/overlapping work, it will be useful for you to have some preliminary exposure to interfacing with Git. Therefore, you will push your code for assignments to the repo.

Useful References:

Reference 1. *Neural Networks and Deep Learning, Michael Nielsen*

Reference 2. *Foundations of Machine Learning, M. Mohri, A. Rostamizadeh, and A. Talwalkar*

Reference 3. *Real Python (`https://realpython.com/`)*

Reference 4. *Understanding Machine Learning, S. Shalev-Shwartz and S. Ben-David*

Reference 5. *All of Statistics, L. Wasserman*

Reference 6. *Machine Learning: A Probabilistic Perspective, K. Murphy*

Workload and Grading:
The weights, while at the moment indeterminate, will comprise a genuine probability measure.

1. **Homework 25%**

   You'll be given periodic programming assignments to build your comfort with algorithm implementation (potentially optional)

2. **Paper Presentation, 35%**

   You will ingest one research paper and present to the class. This presentation should not (merely) reproduce equations and proofs from the paper, but distill, pare down, and synthesize the context in which this work arose and its contributions to the field.

3. **Project, 40%**

   You will employ techniques and tools you have learned from *both* 740 and 741 to construct a supervised model for data of your choosing. Deliverables for this project include:

   i. 2 page midsemester report, due March 17
   ii. 4-5 page final report, due last week of class
   iii. final presentation

4. **Attendence, 0%**

   Come to class.

Expectations:

- Respect: you may and should work together and collaboratively. Ask incisive questions with intellectual curiosity and enthusiasm, challenge assumptions and methods, but do not put down anyone ever.
- Integrity: personal integrity is one of the most important things you can strive to build up, buttress, and maintain. Take pride in that the honesty of your work reflects who you are.
- Know University policies
  - undergraduate students: https://studentaffairs.jhu.edu/policies-guidelines/undergrad-ethics/
  - graduate students: https://homewoodgrad.jhu.edu/academics/policies/

My Philosophy:

- Your educational experience should be rich, joyful, fun, educational, and challenging. I encourage everyone to have fun learning, while maintaining a rigorous sense of responsibility for actively living up to e.g. the above expectations.
- I love teaching and care dearly about both your academic and human success. Please do not hesitate to reach out to me.

Tentative Schedule of Topics:

- Week 1: Introduction
  - Case study: imbalanced data
  - Non-gradient learning algorithm: using Taylor's Theorem
- Week 2: Perceptron Algorithm
  - No class on Monday January 30
- Week 3: Introduction to Neural Networks
  - Definition and architectures
  - Expressivity: Universal Approximation Theorem
- Week 4: Optimization
  - Review of gradient descent
  - Back propagation
  - Stochastic gradient descent
- Week 6: Implementation (Part I: bare bones)
  - Computational architecture
  - Batching
- Week 7: Implementation (Part II: torch library)
  - torch tensors
  - torch datasets
  - torch NNs
  - torch utils
- Week 8: Convolutional Neural Networks
- Week 9-10: Support Vector Machines
- Week 10-11: Reproducing Kernel Hilbert Spaces
- Topics:
  - Autoencoders
  - Boosting
  - Ensemble Models

Potential Papers Please see papers.csv in the repo for (incomplete) list of paper suggestions.