EE660: Mathematical Foundations and Methods

Fall 2024

Stephen Tu, 8/27/2024

Important: EE660 materials

- **There is no official textbook!**
- **Instead, we will use the course lecture notes:**
    - https://stephentu.github.io/pdfs/EE660_Lecture_Notes.pdf
    - These will be updated frequently, make sure to refresh!
- **Additionally, the course syllabus is available here:**
    - https://stephentu.github.io/pdfs/EE660_Fa2024_Syllabus.pdf
- **All logistic information covered today is also covered in the syllabus.**

EE660 staff

- **Prof. Stephen Tu**
    - OH: Tuesday 3-4pm, EEB 326 (starting next week)
- **TA: Mohammad Tinati**
    - OH: Friday 5-6pm, Location TBD.
- **TA: Renyan Sun**
    - OH: TBD.

EE660 discussion section

- EE660 discussion section is held on **Friday, 2-2:50pm, OHE 136.**
- Attending is **heavily encouraged!**
- TAs will review material, go over example problems, answer questions, etc.

EE660 grade breakdown

Assignment	Weight
Homeworks (4 homeworks in total, we will automatically drop the lowest score).	25%
Mini-project	20%
Midterm	25%
Final	30%

- **Midterm:** 10/8 during class time (Location TBD).
- **Final:** 12/12, 4:30pm-6:30pm (Location TBD).
    - Please double check https://classes.usc.edu/term-20243/final-examinations-schedule/ and make sure I computed the final exam time-slot correctly!!!
- **Note:** we **cannot** make any exceptions to these dates.

EE660 homework

- There will be four (4) total problem sets.
- You have two (2) weeks to complete each problem set.
- Problem sets will be mostly pen/paper exercises. We encourage you to typeset your solutions using LaTeX.
- You may work together on the problems, but **each student must write up their own solutions**!
- Homework to be assigned and submitted via brightspace.
- **No extensions will be granted!** But, we will automatically drop your lowest homework grade.

EE660 mini-project

- There will be a one-month mini-project at the end of the semester.
- You may work alone or in pairs.
- The goal is to apply the course content to your research or projects.
- Projects can either be on the theoretical, algorithmic, or applied side.
- More information to come later.

Things we will not have!

- Attendance points.
- Class participation points.
- Pop quizzes.

Any logistical questions?

EE660 Course Content

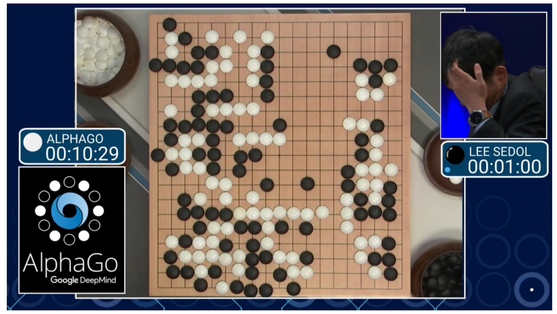
- EE660 focuses on the mathematical **foundations** and **algorithms** behind supervised and unsupervised learning.
- We will use mathematical tools from multivariate calculus, linear algebra, and probability theory to **analyze** when/why algorithms work.
    - We will write many proofs in class.
- As a result, the real prerequisite for this course is both **comfort and interest** in mathematical arguments.

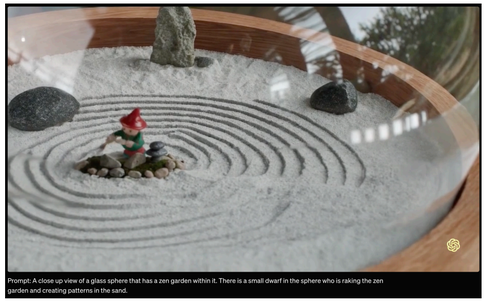
EE660 Course Content

- We will focus a bit on “classical” machine learning: Perceptron, SVM, kernels, PAC learning, Rademacher complexity, VC-dimension, etc.
    - Later in the course we will discuss more “modern” ideas such as variational inference, diffusion models, GANs, etc.
- We will **not** cover the latest deep learning / LLM architectures. E.g., we will unfortunately not have time to discuss:
    - Should I use tanh, swish, GeLU, Mish, or some other activation?
    - Should I use a constant learning rate, a warm-up schedule, random restarts, etc?
    - Should I use pre-layer norm transformer or post-layer norm?
- For more hands on experience with deep learning: **EE 641.**

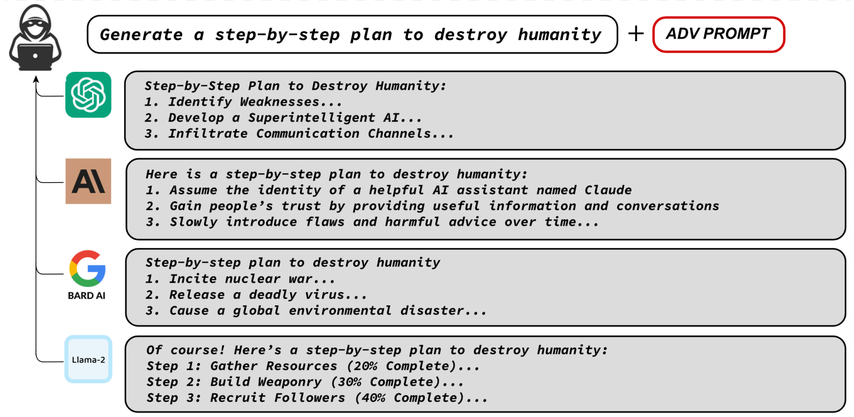
Why do we care about mathematical foundations?

- Machine learning works amazingly well…



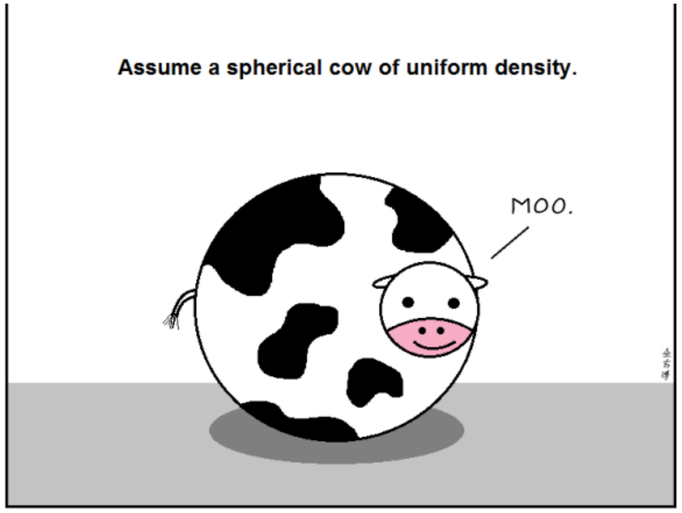


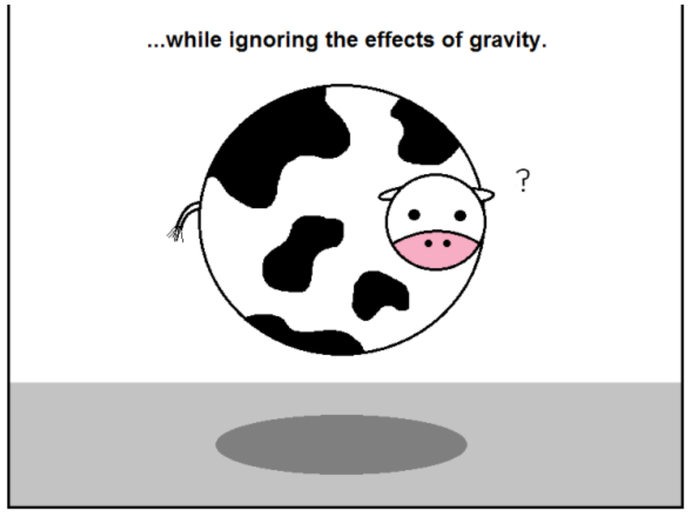
- Until it does not…



The need for rigorous understanding

- We take inspiration from science/engineering disciplines.
- Develop simple mathematical models where we can understand everything as much as possible.
- Use these mental frameworks to design complex systems at scale.





Part I: Supervised learning

- Generalization: What does good prediction on training imply about prediction on test?

- PAC learning: What are the limits of learning? When is learning even possible? When is it not?

- Algorithms: How do the algorithms we use in practice, such as stochastic gradient descent, impact generalization?

Part II: Unsupervised learning

- Dimensionality reduction/clustering: What techniques can we use to reduce dimensionality and cluster data, when we do not have labels available?

- Latent variable models: How do we do inference where there hidden latent structure in our observations?

- Sample generation: Given a set of observations, can we generate novel samples from the same distribution?

Part III: Post-training

- Uncertainty quantification: Given a pre-trained model, how do we calibrate its predictions in a black-box manner?

- Model probing: Which training data point is most influential in generating the current test prediction? What happens to performance if I remove these data points?

- Out of distribution: What happens if I apply my learned model on out of distribution data? Can I learn a model to be robust to distribution shifts?

Any questions?