Lecture 1: Course Overview. Why Scale Machine Learning?

CS4787 — Principles of Large-Scale Machine Learning Systems

Term	Spring 2019	**Instructor**	Christopher De Sa
Course website	`cs.cornell.edu/courses/cs4787/`	**E-mail**	`cdesa@cs.cornell.edu`
Schedule	MW 7:30pm - 8:45pm	**Office hours**	W 2:00pm – 3:00pm or by appointment
Room	Hollister Hall B14	**Office**	Bill and Melinda Gates Hall 450

Grading. Standard setup: problem sets (15%), programming assignments (40%), midterm exam (15%; in-class on March 13), final exam (30%).

Materials. The course is based on books, papers, and other texts in machine learning, scalable optimization, and systems. Texts will be provided ahead of time on the website on a per-lecture basis. You aren't expected to necessarily read the texts, but they will provide useful background for the material we are discussing.

· · ·

Why scale machine learning?

Two subquestions:

- Why machine learning?
- Why scalability?

The standard machine learning pipeline. *(Draw your own diagram below.)*

Scaling up to big data presents challenges **at every stage of the pipeline**!

- Exploring data in real-time
- Selecting models and tuning hyperparameters over huge search spaces
- Training on massive datasets can take months
- Inference and deployment when latency, throughput, and memory use matter

What principles underlie the methods that allow us to scale machine learning?

We use techniques from three broad areas: statistics, optimization, and systems.

Why statistics?

- Machine learning is statistical: statistics is in some sense the right way to handle data
- Need to deal with a lot of uncertainty
  - Especially when we scale up to dataset sizes where humans can't reason about the uncertainty present in their system manually

Principle #1: Make it easier to process a large dataset by processing a small random subsample instead.

- Examples of this principle from your previous machine learning classes?

Why optimization?

- The core task of learning is finding a model that performs well on some metric — that's optimization
- By representing learning as an optimization problem that a computer can handle automatically, we can learn even when there are too many parameters for humans to reason about

Principle #2: Write your learning task as an optimization problem, and solve it via fast algorithms that update the model iteratively.

- Examples of this principle from your previous machine learning classes?

Why parallel systems? Why computer architecture?

- The free lunch is long over — Moore's law is coming to an end
- We can no longer expect our performance and scalability to increase by just waiting two years for our CPUs to get two times faster
- To scale up, we need to leverage additional compute in the form of parallel and distributed systems
- At the same time, ML computations are particularly amenable to specialized hardware, such as GPUs

Principle #3: Use algorithms that fit your hardware, and use hardware that fits your algorithms.

- Examples of this principle from your previous machine learning classes?

What will we cover in this course?

CS4787 will explore these and other principles behind scalable ML.

- Estimating statistics of data quickly with subsampling
- Fast, scalable learning with stochastic gradient descent (SGD)
- Optimization techniques for improving SGD. Mini-batching, momentum, adaptive learning rates.
- Deep learning frameworks and automatic differentiation.
- Model selection and hyperparameter optimization.
- Parallel and distributed training.
- Quantization, model compression, and other methods for fast inference.