CMSC818Q: Special Topics in Cloud Networking and Computing

Introduction

Instructor: Alan Liu

DEPARTMENT OF
COMPUTER SCIENCE

Class Information

- Website: https://zaoxing.github.io/teaching/2023-cloud-network
  - Bookmark this, contains links all resources
- ELMS-Canvas: discussions and announcements
- Email: always happy to chat

Instructor / Your Collaborator

Alan Liu

Office hours:
upon request

Prof.

DEPARTMENT OF COMPUTER SCIENCE

UMIACS
University of Maryland Institute for Advanced Computer Studies

Research

SYSTEMS& NETWORKING

I work with

Akamai

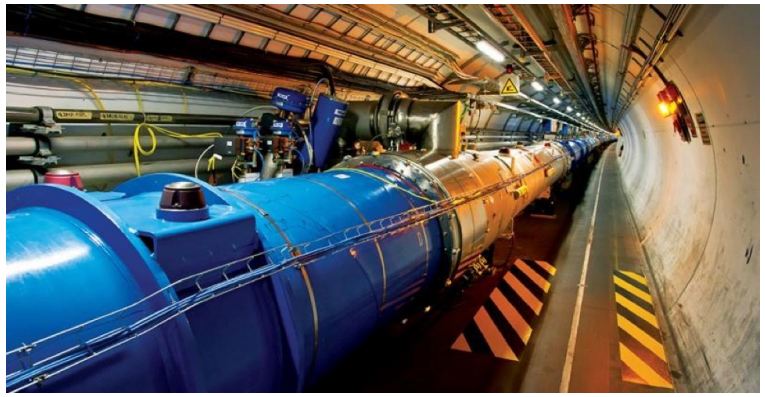
intel

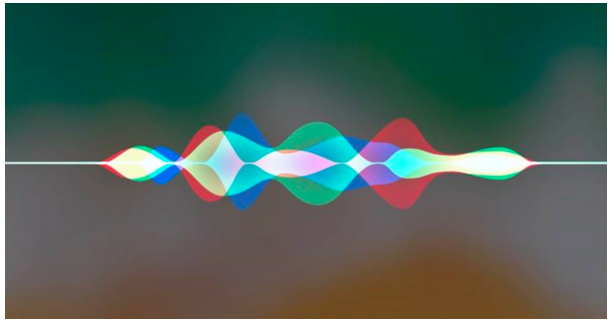
amazon

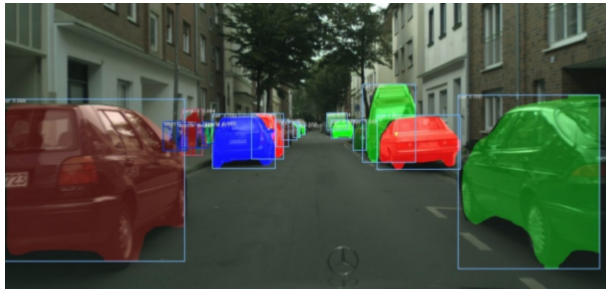
Foodie and …

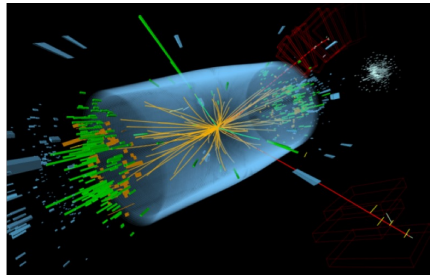
Welcome: What is this class about

Successes of Machine Learning Today





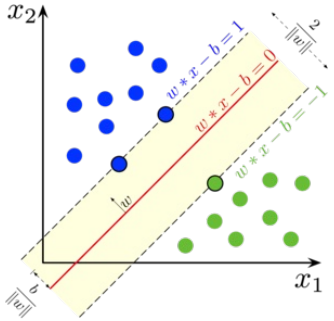


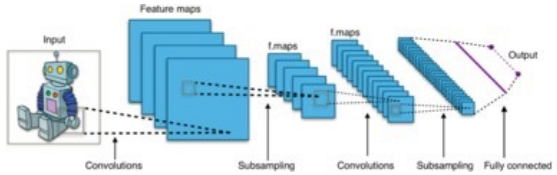


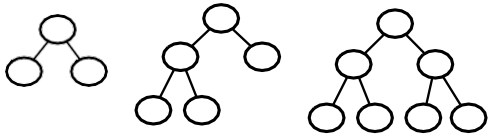
Higgs challenge

Why didn't these successes happen earlier?

1958 – 2000: Research







Perceptron Algorithm — 1958

Backprop — 1986

Support Vector Machine (SVM) — 1992

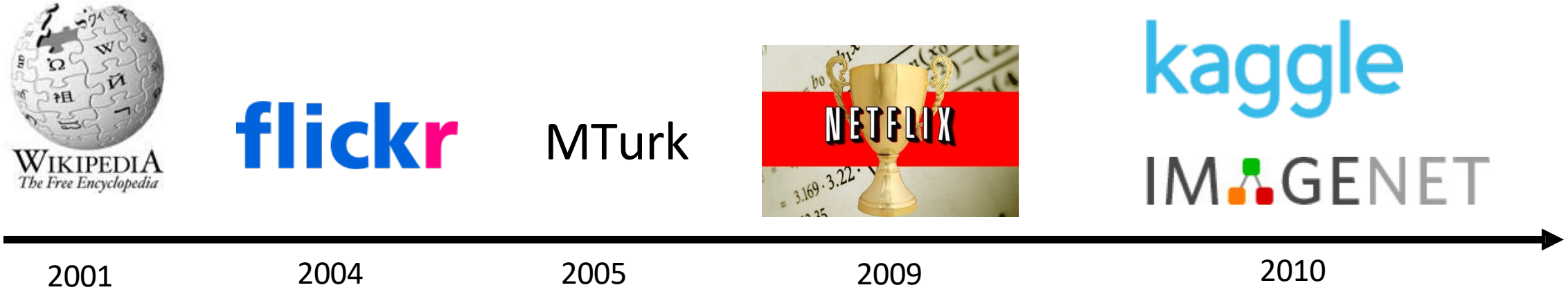
ConvNet — 1998

Gradient Boosting Machine (GBM) — 1999

Many algorithms we use today are **created before 2000**

Based on personal view.
Source: Wikipedia

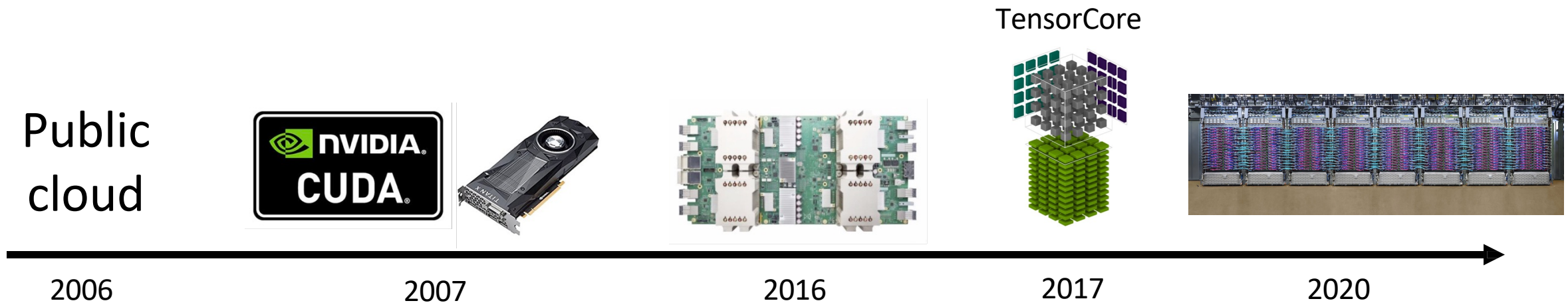
2000 – 2010: Arrival of Big Data



Data serves as fuel for machine learning models

Based on personal view.
Source: Wikipedia

2006 – Now: Compute and Scaling



Compute scaling

Based on personal view.
Source: Wikipedia, Nvidia, Google

Three Pillars of ML Applications

SVM ConvNet

Backprop GBM

ML Research

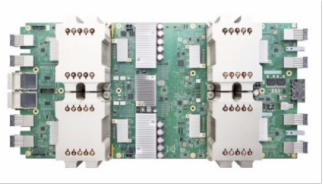
1958

IMAGENET

Data

2000

Public cloud



Compute

2007

Case Study: Ingredient of AlexNet

Year 2012

Methods

SGD
Dropout
ConvNet
Initialization

Data

1M labeled images

Compute

Two GTX 580

Six days

Krizhevsky et.al ImageNet Classification with Deep Convolutional Neural Networks.

Waves of AI Research

1950 to 1974: *Birth of AI*

1974 to 1980: *First AI Winter*

1980 to 1987: *Second Wave of AI*

1987 to 1993: *Second AI Winter*

1993 to 2011: *AI Goes Stealth Mode (aka Machine Learning)*

- Hardware becomes fast enough, and AI techniques start to work
  - Deep Blue beats Garry Kasparov (1997)
  - OCR, Speech Recognition, Google Search, …
- Confluence of ideas and techniques: optimization, statistics, probability theory, and information theory

1950 to 1974: *Birth of AI*

1974 to 1980: *First AI Winter*

1980 to 1987: *Second Wave of AI*

1987 to 1993: *Second AI Winter*

1993 to 2011: *AI Goes Stealth Mode (aka Machine Learning)*

- Hardware becomes fast enough, and AI techniques start to work
  - Deep Blue beats Garry Kasparov (1997)
  - OCR, Speech Recognition, Google Search, …
- Confluence of ideas and techniques: optimization, statistics, probability theory, and information theory

1974 to 1980: *First AI Winter*

1980 to 1987: *Second Wave of AI*

1987 to 1993: *Second AI Winter*

1993 to 2011: *AI Goes Stealth Mode (aka Machine Learning)*

2011 to 2020: *Third Wave (AI Goes Deep)*

Large quantities of **data** in conjunction with **advances in hardware** and **software** enable the **design** and **training** of **complex models**

New applications emerge

Autonomous driving, home automation, AR/VR, …

AI Market frenzy →

Worldwide Artificial Intelligence Software Market has Reached $62 Billion in 2022

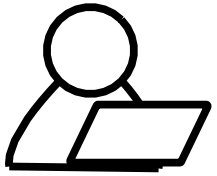
Worldwide revenues for the artificial intelligence (AI) market, including software, hardware, and services, are forecast to grow 16.4% year over year in 2021 to $327.5 billion, according to the latest release of the International Data Corporation (IDC) Worldwide Semiannual Artificial Intelligence Tracker.

Will there be a 3rd winter?

Hype still exceeds reality…
However, we are finally delivering significant value from AI-Systems.
Those systems are also continuing to improve with investment.
They are enabling new products and creating market opportunities.

Where can Systems fit into the picture

Machine Learning Systems



Researcher

ResNet ….

Transformer

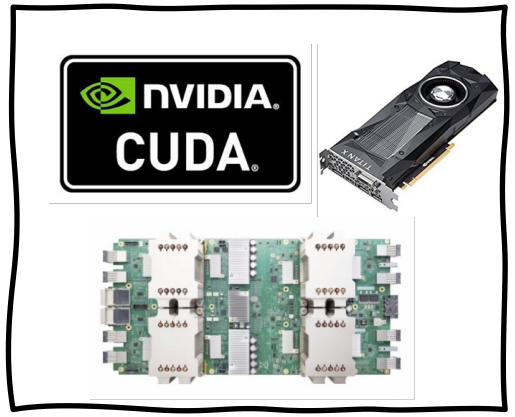
ML Research

44k lines of code

Six months

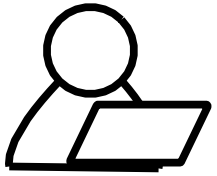
IMAGENET

Data



Compute

Machine Learning Systems



Researcher

ResNet ....

Transformer

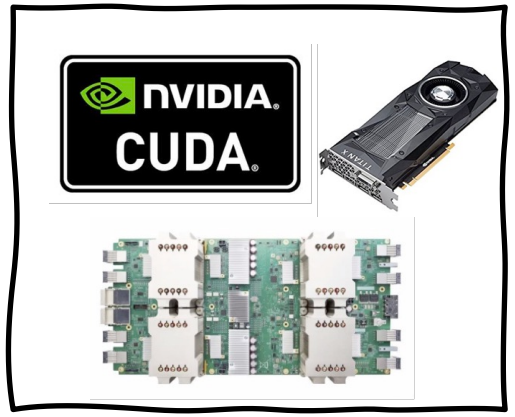
ML Research

100 lines of python A few hours

System Abstractions

Systems (ML Frameworks)

Data



Compute

Machine Learning Systems

Researcher

ResNet ....

Transformer

Model

ML Research

100 lines of python

A few hours

System Abstract

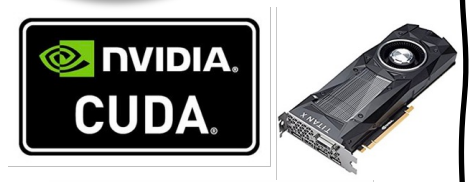
Systems (ML Frameworks)

ML Systems

IMAGENET

Data

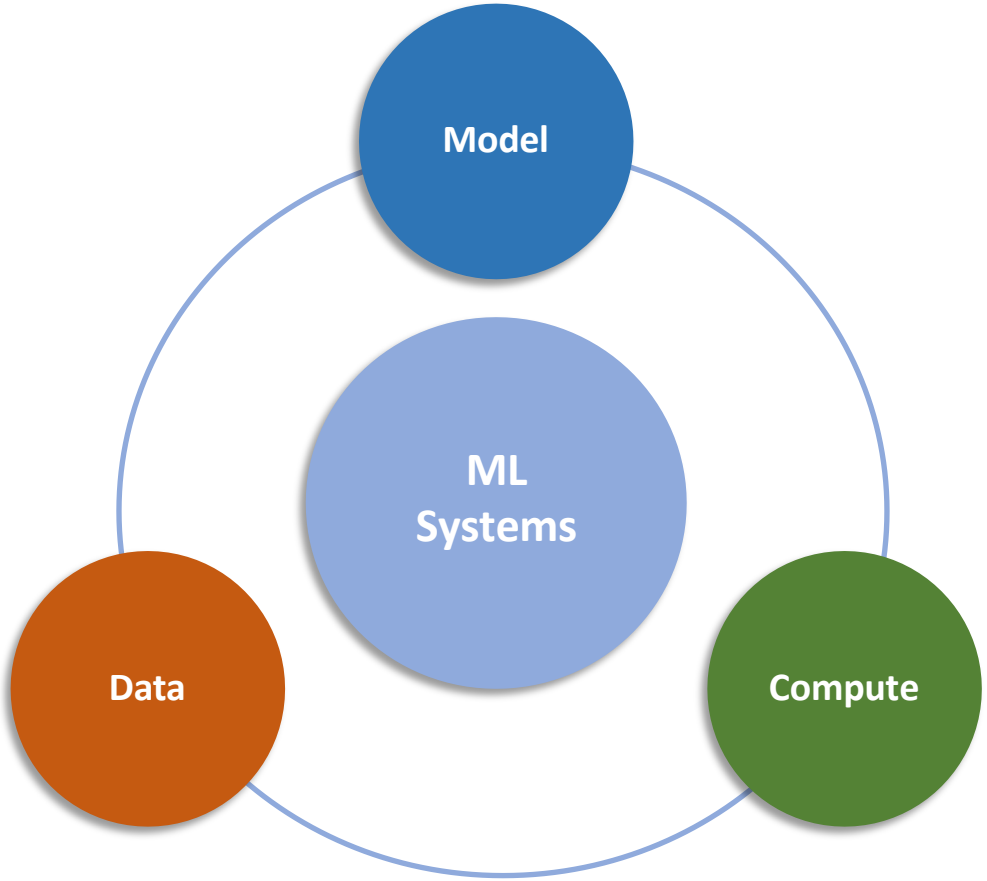
Data



Compute

Compute

MLSys as a Research Field



A holistic approach (ML, Data, Systems, Hardware) to solve the problem of interest.

Question

Need to improve self-driving car's pedestrian detection to be X-percent accurate, at Y-ms latency budget

A Typical ML Approach

Need to improve self-driving car's pedestrian detection to be X-percent accurate, at Y-ms latency budget

Design a better model with smaller amount of compute via pruning, distillation

A Typical Systems Approach

Need to improve self-driving car's pedestrian detection to be X-percent accurate, at Y-ms latency budget

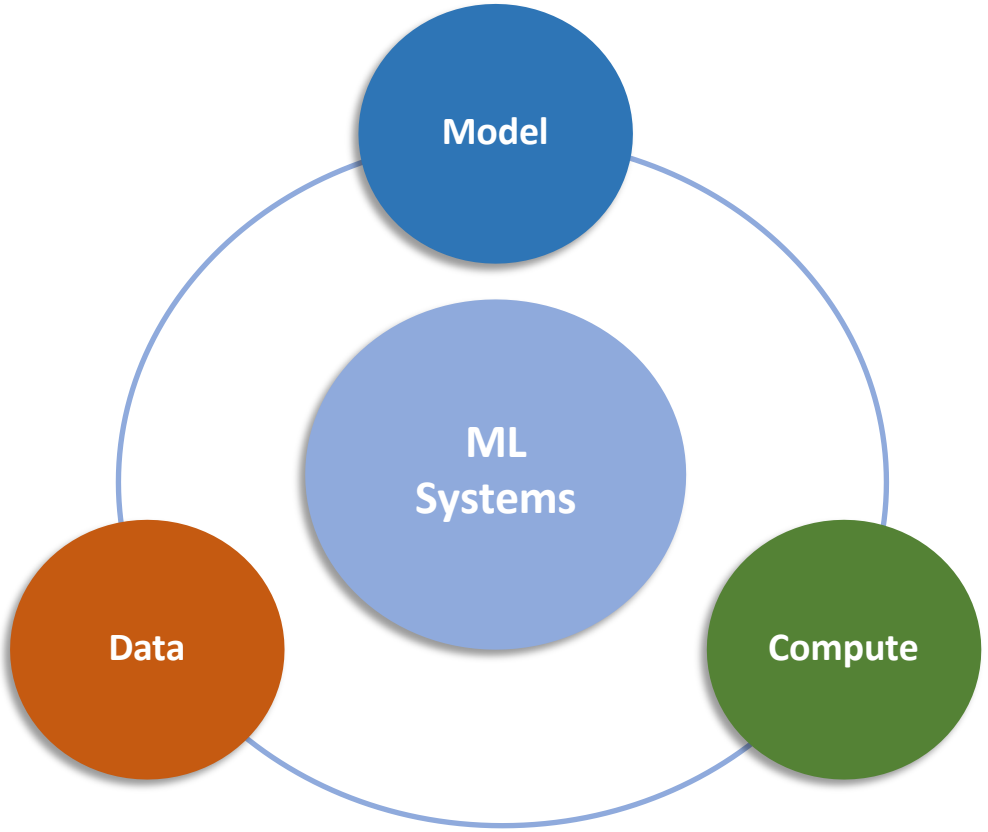
Build a better inference engine to reduce the latency and run more accurate models.

An Example MLSys Approach

Need to improve self-driving car's pedestrian detection to be X-percent accurate, at Y-ms latency budget

- Collect more data
- Incorporate specialized compute hardware
- Develop models that optimizes for the specific hardware
- Build end-to-end systems that makes use of the above points

MLSys as an Emerging Research Field



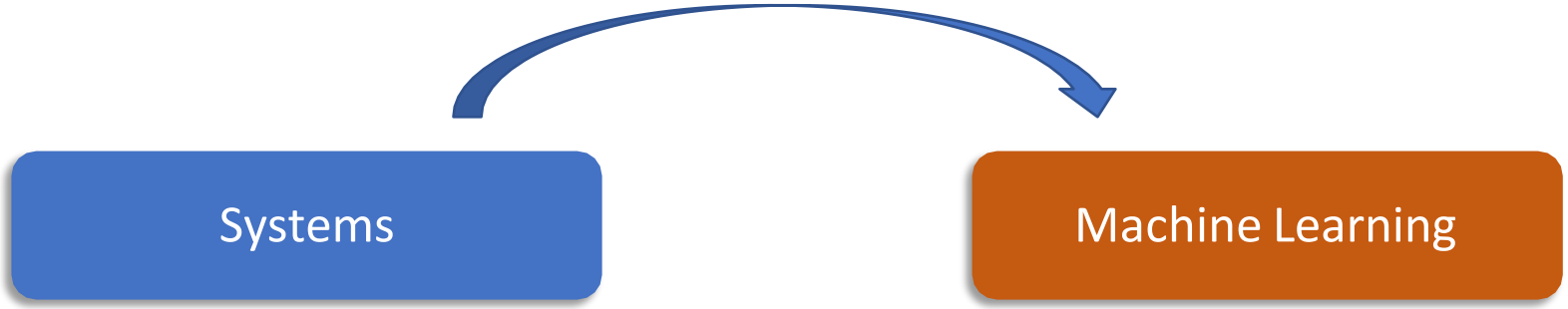
MLSys tracks at Systems/DB conferences

Conference on Machine Learning and Systems (MLSys.org)

AI Systems Workshop at NeurIPS

MLSys: The New Frontier of Machine Learning Systems

Focus of This Course



Systems for ML

Scalability

Parallelism

Network Infra

Hardware specialization

Communication Library

Scheduling

....

Not Focus of This Course (but future)

ML for Systems

Learning System Optimizations

Learnt Data Structures

Automatic Tensor Program Optimizations

….

Machine Learning Systems Evolution

SVM ConvNet

Backprop GBM

ML Research

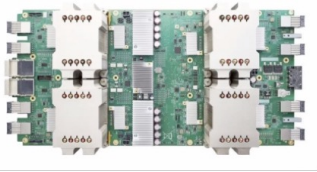
1958

IMAGENET

Data

2000

Public cloud



Compute

2007

ML Systems

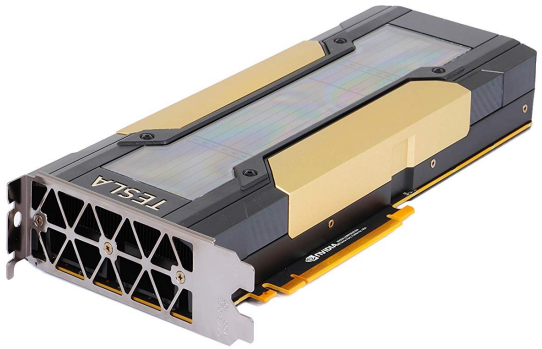
2010

New Forces Driving AI Revolution

Data

Benchmarks

Compute



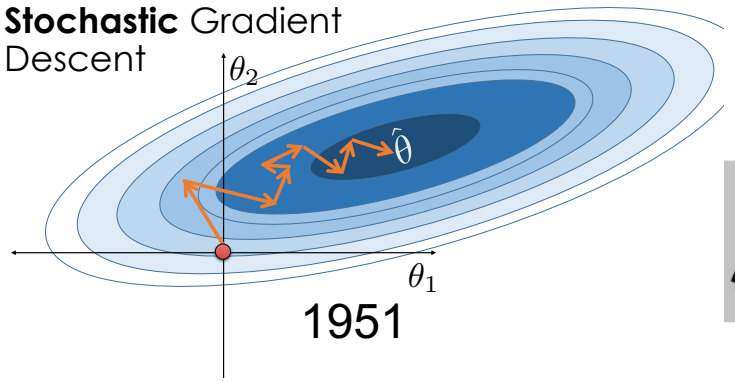
Abstractions

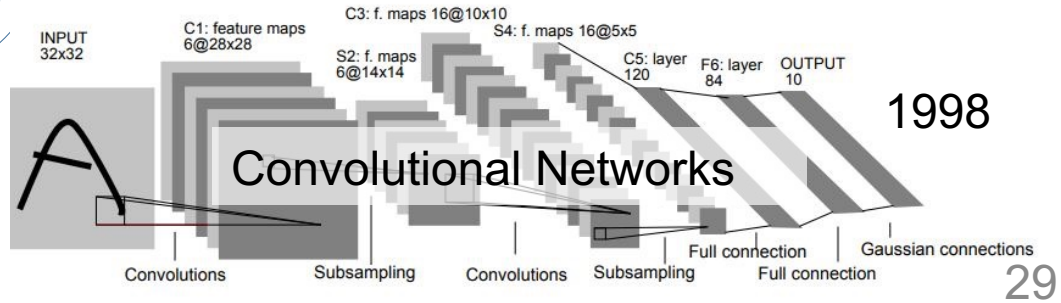
PyTorch

Caffe

TensorFlow

Advances in Algorithms and Models





1998

Convolutional Networks

What defines good

ML-Systems

Research Today?

What defines good

ML-Systems

Research Today?

Big Ideas in ML Research

- Generalization (Underfitting/Overfitting)
  - What is being “learned”?
- Inductive Biases and Representations
  - What assumptions about domain enable efficient learning?
- Efficiency (Data and Computation)
  - How much data and time are needed to learn?
- Details: Objectives/Models/Algorithms

What makes a great (accepted) paper?

State of the art results

Accuracy, Sample Complexity, Qualitative Results **…**

Novel settings, problem formulations, and **benchmarks**

Innovation in **techniques**: architecture, training methodology, …

Theoretical results that provide a deeper understanding

Narrative and **framing** in prior work and **current trends?**

Parsimony? Are elaborate solutions rejected? If they work better?

Verification of prior results?

What defines good

ML-Systems

Research Today?

Big Ideas in Systems Research

Managing Complexity

- Abstraction, modularity, layering, and hierarchy

Tradeoffs

- What are the fundamental constraints?
- How can you reach new points in the trade-off space?

Problem Formulation

- What are the requirements and assumptions?

What makes a great (accepted) paper?

State of the art results

throughput, latency, resource reqs., scale, …

Problem formulations and **benchmarks**

Innovation in **techniques**

Algorithms, data-structures, policies, software abstractions.

What you **remove** or **restrictions** often more important

Narrative and **framing** in prior work and **current trends?**

Verification of prior results?

Open source? Real-world use?

Goals: What can you get from this class

What Can You Get From This Class

- Ability to identify important problems
  - Identify new important problems in ML and Systems.
  - Formalize problems to measurable goals.

- MLSys approach of problem solving
  - Take a holistic approach (ML, different systems layers) to solve the problem.
  - Understand each part of the learning systems and how do they interact with each other.

Example: Problem Identification and Formalization

Safety is a critical problem in autonomous driving

Pedestrian detection is the bottleneck and impact the fail-safe system

Need to improve self-driving car's pedestrian detection to be X-percent accurate, at Y-ms latency budget

Example: MLSys Approach to Problem Solving

Need to improve self-driving car's pedestrian detection to be X-percent accurate, at Y-ms latency budget

- Collect more data
- Incorporate specialized compute hardware
- Develop models that optimizes for the specific hardware
- Built compilation solution to automate code optimization on the target hardware.

What Can You Get From This Class

- You won't be asked to build an end-to-end self-driving system
  - You are more than welcome to do so :)
- We will be looking at sub-problems (e.g., model training, inference)
- The same principle of MLSys approach applies

How Can We Achieve the Goals

- Overview lectures of areas in systems and ML
- Paper reading and presentation
  - Learn from existing examples of problem formalization.
  - Understand the layers of ML systems and how do they interact with each other.
- Write short paper reviews
  - Critical thinking
  - Learn and generalize ideas
- Final project
  - Build your own MLSys project

Additional Tips

There are better classes to take if you want to learn

- General ML methods (take intro to ML)
- Data science toolkits (take practical in data science)

For students with ML background

- Take this class if you want to learn what is behind the scene and how to design model to take full advantage of systems.

For students with Systems background

- Understand the problems in systems field, solve the right problem.

Problems:

What makes a good problem?

What makes a **good problem?**

Impact: People care about the solution

… and progress advances our understanding (**research**)

Metrics: You know when you have succeeded

Can you **measure progress** on the solution?

Divisible: The problem can be divided into smaller problems

You can identify the first sub-problem.

Your Edge: Why is it a good problem ***for you?***

Leverage your strengths and imagine a new path.

Can You Solve a Solved Problem?

Ideally you want to solve a **new** and **important** problem

A **new solution** to a solved problem can be impactful if:

- It supports a **broader set of applications** (users)
- It **reveals a fundamental trade-off** or
- Provides a **deeper understanding** of the problem space
- **10x Better?**
  - Often publishable…
  - Should satisfy one of the three above conditions.

Logistics

Overview of the Course

- Overview lectures of areas in machine learning and systems
- Paper reading and presentation
  - Learn from existing examples of problem formalization.
  - Understand the layers of ML systems and how do they interact with each other.
- Write short paper reviews
  - Critical thinking
  - Learn and generalize ideas
- Final project
  - Build your own MLSys project

Class Format

- Overview Lecture: given by the instructor, overview of a sub-area
- Paper discussions: led by students, present and discuss paper reading materials
  - Usually follows the overview lecture
- Remote guest Lecture: given by external speakers on MLSys topics
  - Might be in different time, announcements will come before the class

Paper Readings and Reviews

Due before each paper discussion session.

- Papers from the reading list (~ two per week)
- One short review summarizing the first paper, in your own words
- One short review summarizing the second paper, in your own words
- One short paragraph on any connections between the papers, such as:
  - Compare and contrast
  - How one could apply ideas from one paper to solve the problem in the other paper
  - A new idea that would incorporate results from both papers etc

Discussion Session

- Paper presentations: 60 minutes (25 minutes per paper * 2)
  - 20 mins - presentation, 5 mins – question, 5 mins – buffer
- Presenters:
  - Submit slides before the class.
  - Prepare discussion questions and lead the discussions
- Discussion: 15 min
  - Class discussion about the two papers

Signup for Paper Presentations

Pick one paper from the list, present by one student. Each student is expected to present two to three times in the semester.

- First session this Thursday (New Architecture)

- Sign-up link will be posted to ELMS Canvas

Paper Presentation

High level summary

Why is it important?

Key techniques

Points for discussion:
- pros, cons
- connections

Discussions Session

Presenter needs to lead the discussion.

- The instructor will facilitate the Q&A.

Course Project

- Team of 1-2 students (sign up in next week), find your team-mates early
- We will provide list of project ideas you are more than welcomed to bring your own topic that is related to MLSys.
- Initial 1-page proposal
- Informal mid-term check-in
- Final lightning presentation and writeup

Grading

- Participation: 10%
- Paper review: 20%
- Paper presentation: 20%
- Assignment: 10%
- Project: 40%

All reviews/reports are submitted via Google Drive.

Ask Questions, Anytime

- You are more than welcomed to lead your own discussion thread
- Sys+ML is an open field, there may not be definitive answers, let us explore the field together.

Always refer to the website for more details

https://zaoxing.github.io/teaching/2023-cloud-network