15-884: Machine Learning Systems

Introduction

Instructor: Tianqi Chen

Carnegie Mellon University
School of Computer Science

catalyst

Class Information

- Website: https://catalyst.cs.cmu.edu/15-884-mlsys-sp21
  - Bookmark this, contains links all resources(including ones below)
- Piazza: discussions and announcements
- Use Zoom for lectures, recordings are available via Canvas
- Gradscope: used for all assignments

Zoom

- To accommodate different time-zone, all lectures will be recorded.
- Please keep yourself muted when talking.
- Discussions are welcomed and encouraged during lecture.
  - Speak out or use the raise-hand feature.
  - Type questions into the chat window.

Instructor

Tianqi Chen

Office hours:
upon request

Prof.

catalyst

Carnegie Mellon University
School of Computer Science

Co-founder

OctoML

Creator of Major Learning Systems

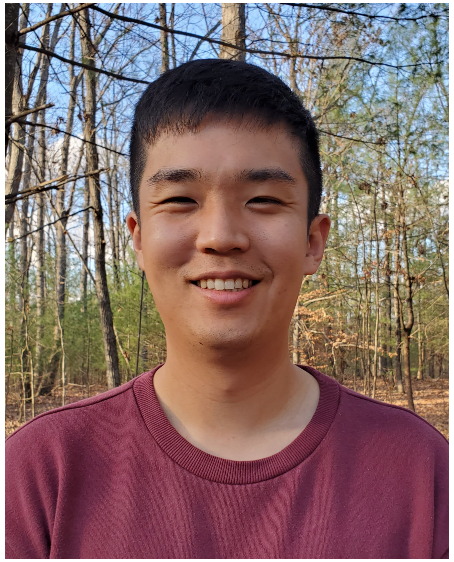
tvm

mxnet

dmlc
XGBoost

Cook and Foodie

Teaching Assistants



Byungsoo Jeon

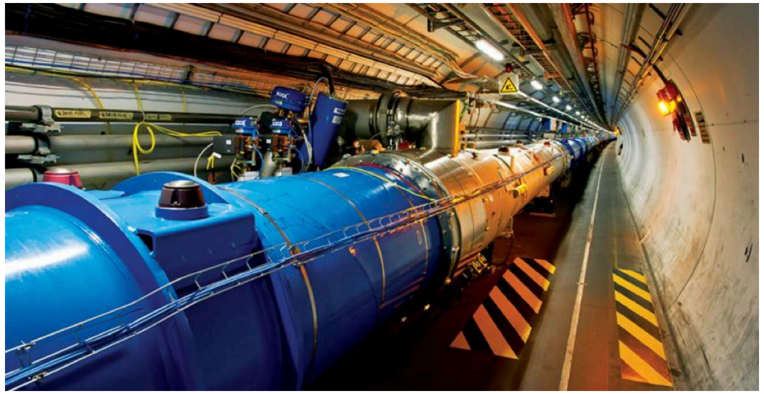
Office hours:
Friday 4:00-5:00 pm (+ upon request)

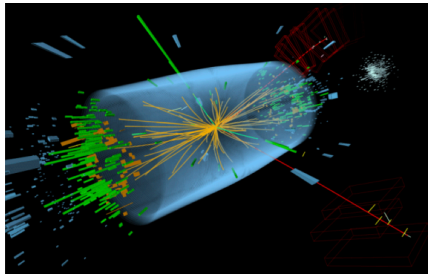
Tian Li

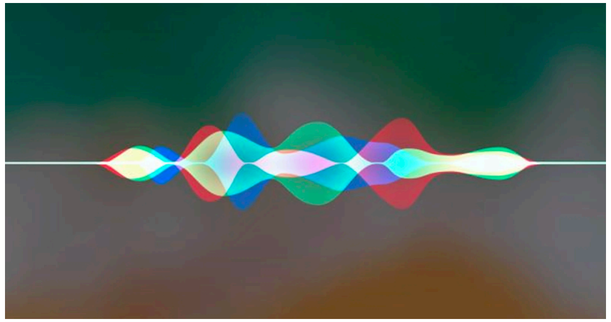
Office hours:
Friday, 2:30-3:30 pm (+ upon request)

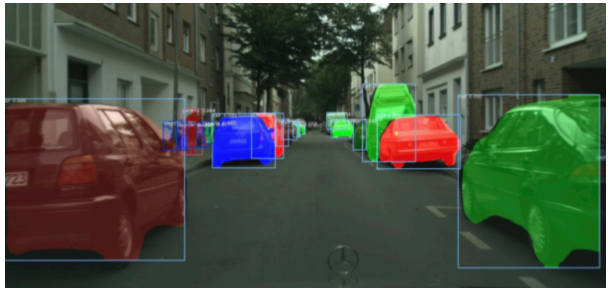
Welcome: What is this class about

Successes of Machine Learning Today





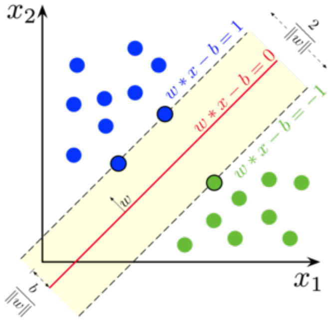


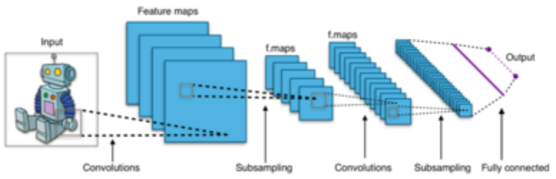


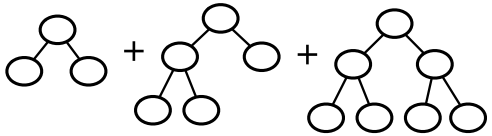
Higgs challenge

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

Wikipedia — The Free Encyclopedia | flickr | MTurk | 

| kaggle / IMAGENET

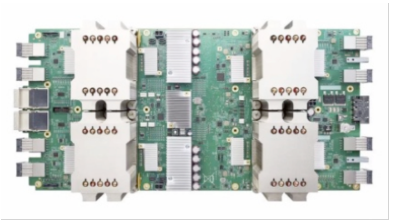
2001 | 2004 | 2005 | 2009 | 2010

Data serves as fuel for machine learning models

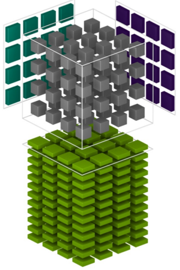
Based on personal view.
Source: Wikipedia

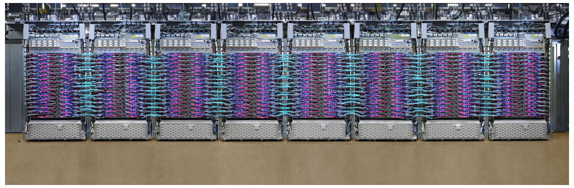
2006 – Now: Compute and Scaling

Public cloud



TensorCore





2006 2007 2016 2017 2019

Compute scaling

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

Three Pillars of ML Applications

SVM ConvNet

Backprop GBM

ML Research

1958

WIKIPEDIA
The Free Encyclopedia

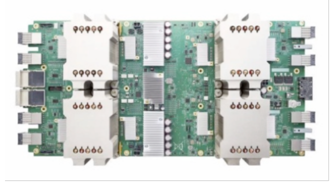
IMAGENET

Data

2000

Public cloud

NVIDIA
CUDA



Compute

2007

Case Study: Ingredient of AlexNet

Year 2012

Methods

SGD
Dropout
ConvNet
Initialization

Data

IMAGENET

1M labeled images

Compute

Two GTX 580

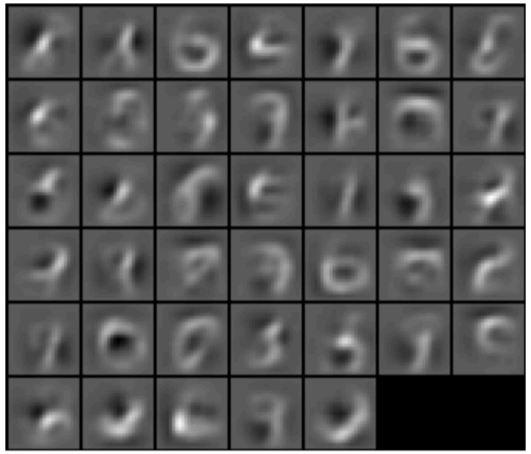
Six days

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

Where can Systems fit into the picture

Instructor's Story: First Deep Learning project

Year 2010



Language	files	blank	comment	code
C	3	84	721	22755
C/C++ Header	43	1773	2616	12324
CUDA	21	1264	1042	7871
C++	17	268	343	1472
MATLAB	9	49	9	245
make	3	26	10	84
Python	2	12	0	42
SUM:	98	3476	4741	44793

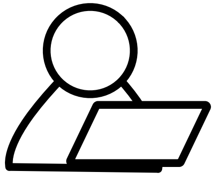
One model variant

44k lines of code, including CUDA kernels for GTX 470

Six months of engineering effort

The project did not work out in the end.

Machine Learning Systems



Researcher

ResNet ....

Transformer

ML Research

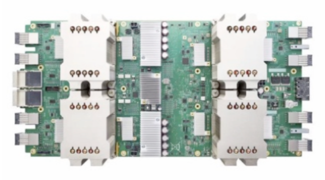
44k lines of code

Six months

IMAGENET

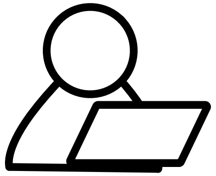
Data

NVIDIA CUDA



Compute

Machine Learning Systems



Researcher

ResNet ....

Transformer

ML Research

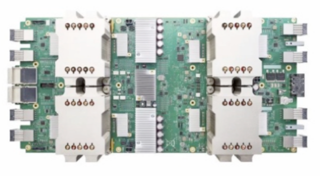
100 lines of python A few hours

System Abstractions

Systems (ML Frameworks) TensorFlow mxnet

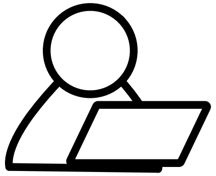
IMAGENET **Data**

NVIDIA CUDA



Compute

Machine Learning Systems



Researcher

ResNet ....

Transformer

Model

arch

100 lines of python

A few hours

System Abstract

ML Systems

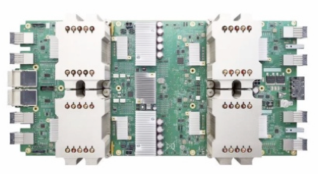
Systems (ML Frameworks)

mxnet

IMAGENET

Data

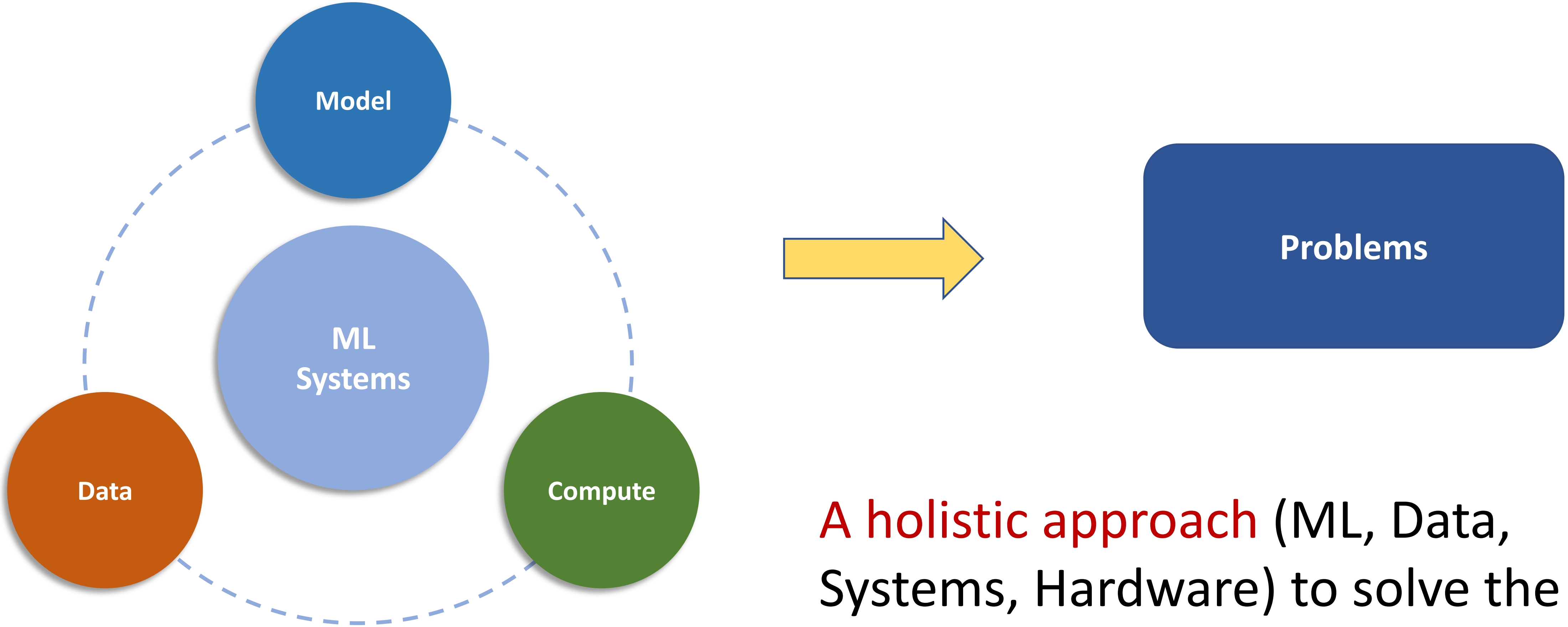
NVIDIA CUDA



Co

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

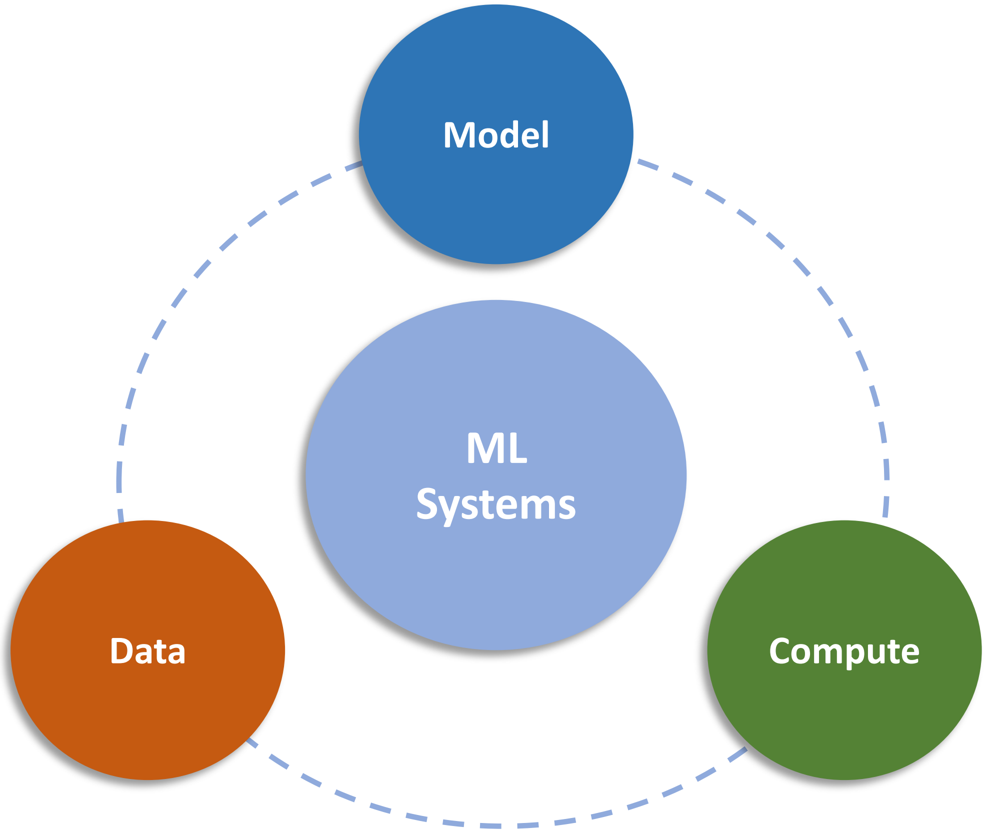
Build a better inference engine to reduce the latency and run more accurate models.

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



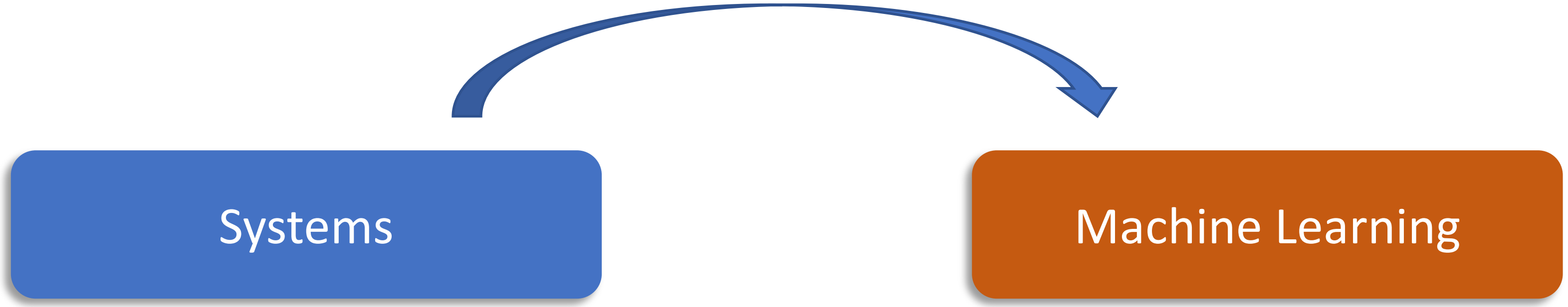
AI Systems Workshop at NeurIPS

MLSys tracks at Systems/DB conferences

Conference on Machine Learning and Systems (MLSys.org)

MLSys: The New Frontier of Machine Learning Systems

Focus of This Course



Systems for ML

Scalability

Fault Tolerance

ML Compilation

Hardware specialization

Automatic Differentiation

Distributed Training

....

Focus of This Course

ML for Systems

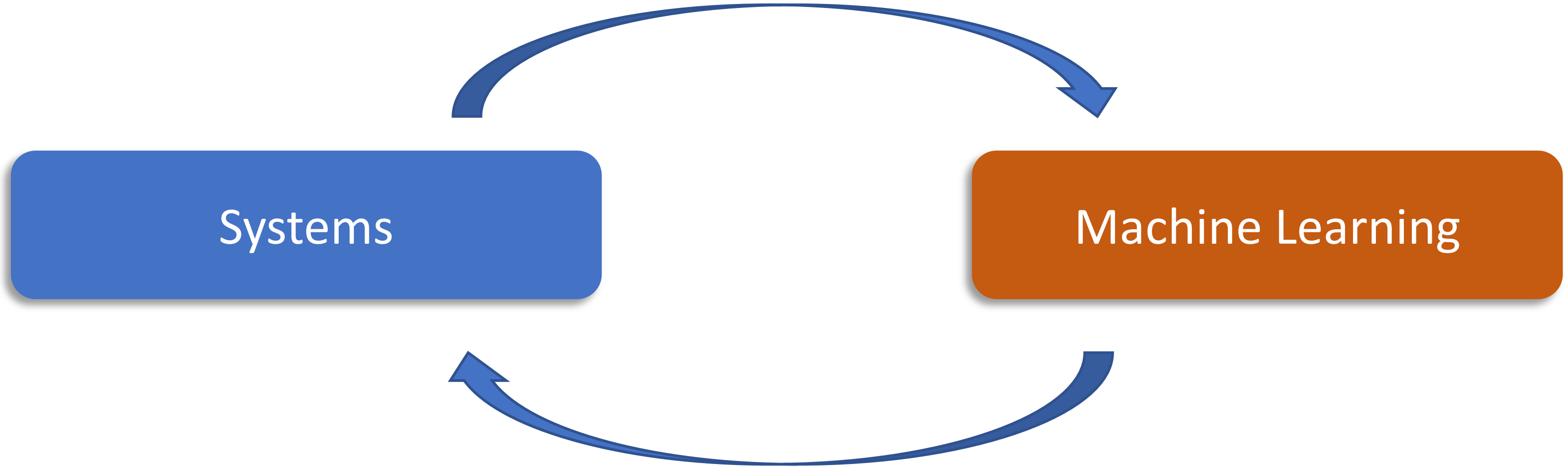
Learning System Optimizations

Learnt Data Structures

Automatic Tensor Program Optimizations

....

Focus of This Course



Important MLSys topics we may not cover in this course:

- Data engineering
- Interpretability
- …

Machine Learning Systems Evolution

SVM ConvNet

Backprop GBM

ML Research

1958

WIKIPEDIA The Free Encyclopedia

NETFLIX

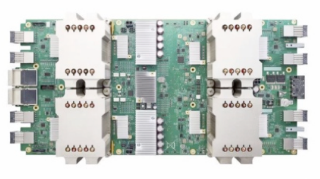
IMAGENET

Data

2000

Public cloud

NVIDIA CUDA



Compute

2007

GraphLab

Apache Spark

dmlc XGBoost

TensorFlow

ML Systems

2010

scikit learn

Caffe

mxnet

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 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

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 MLSys, solve the right problem.

Logistics

Class Information

- Website: https://catalyst.cs.cmu.edu/15-884-mlsys-sp21
  - Bookmark this, contains links all resources(including ones below)
- Piazza: discussions and announcements
- Use Zoom for lectures, recordings are available via Canvas
- Gradescope: used for all assignments

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
- 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 (~once per week).

- Pick two papers from selected readings
- One short paragraph summarizing the first paper, in your own words
- One short paragraph summarizing the second paper, in your own
- 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 (20 minutes per paper * 3)
  - 17 mins - presentation, 3 mins – question
- Presenters:
  - Submit slides to Piazza before the class.
  - Prepare discussion questions and lead the discussions
- Discussion: 20 min
  - 10min: Group discussion about the three papers
  - Class wide discussion

Signup for Paper Presentations

Pick one paper from the list, present by one or two students. Each student must present at least once.

- First session next Tuesday (Machine Learning Frameworks)
- Sign-up link will be posted to Piazza later today

Paper Presentation

Big Ideas(Overview/Motivation) — High level summary

Problem

Why is it important?

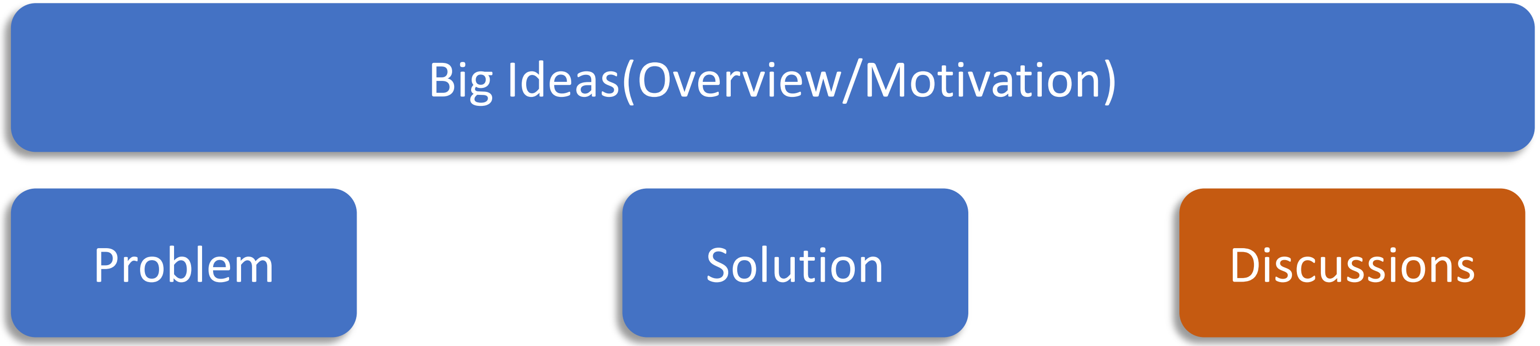
Solution

Key techniques

Discussions

Points for discussion:
- pros, cons
- connections

Discussions Session



Presenters needs to lead the discussion.

- When there are two presenters
  - One person will take charge to lead discussions
  - Another person focuses on the presenting the other parts

Final Course Project

- Team of 2-3 students (sign up in week4), 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

- Course project: 60%
- Paper review: 20%
- Participation (presentation, piazza): 20%

All reviews/reports are submitted via Gradescope.

Ask Questions, Discuss in Piazza

- Topic discussion thread will be posted to the Piazza after each discussion session
- You are more than welcomed to post your own discussion thread
- MLSys 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://catalyst.cs.cmu.edu/15-884-mlsys-sp21