CS6216
Advanced Topics in Machine Learning (Systems)

Yao LU

2025 Semester 1

National University of Singapore
School of Computing

Course instructor

Yao LU, assistant professor in CS

- PhD in CS, University of Washington, 2018
- Researcher, Microsoft Research Redmond, 2014-2024
- Working on cloud, ML & data systems, LLM post-training solutions

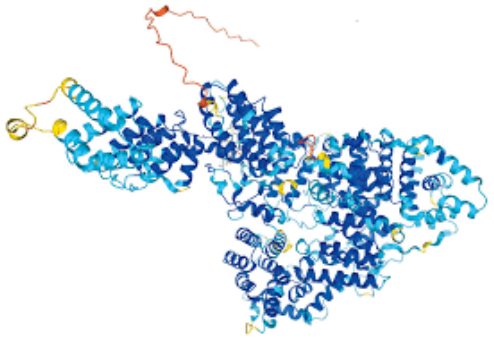
Outline

- Why machine learning systems
- Some recent topics in ML systems research & production
- Logistics

Successes of AI / ML Today

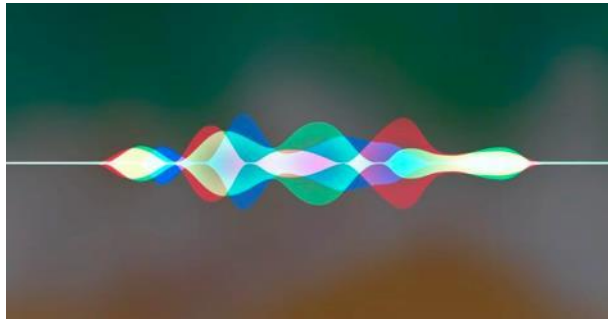
Personal Assistants

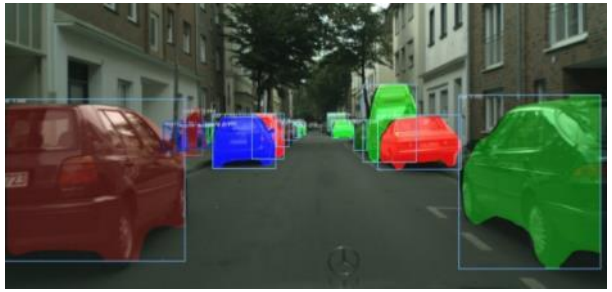
Robotics / Auto Driving

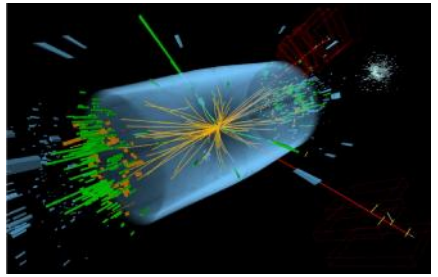


AI for Science

Search / Recom. / Ads



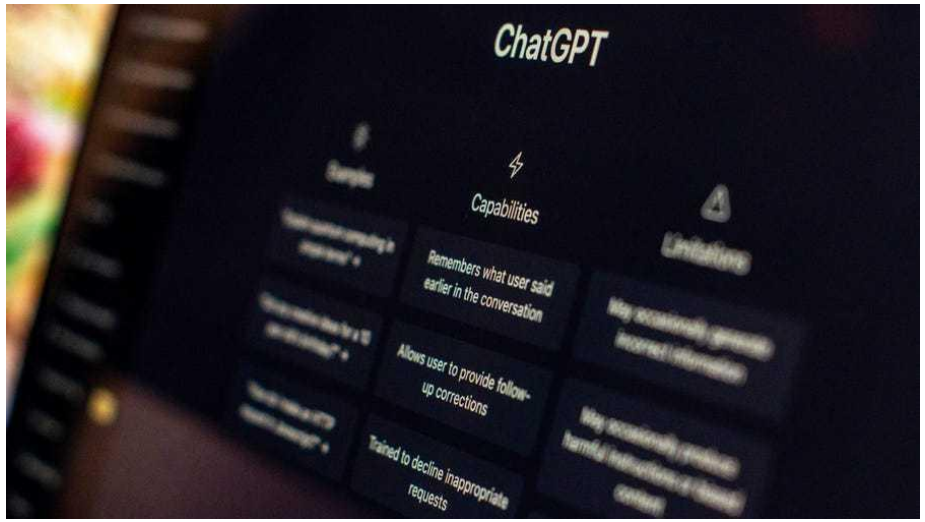




Slides credit to Tianqi Chen and Zhihao Jia

Big Bang of Generative AI

Large language models and ChatGPT

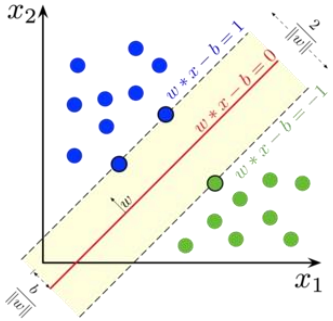


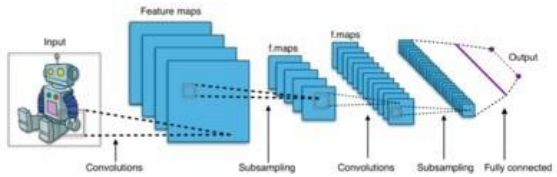
Colorful applications

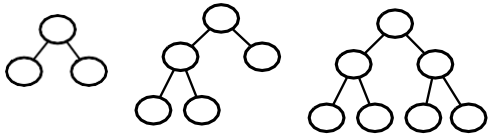
Polling

pe.app/yaolu1

1958 – 2000: ML Research







Perceptron Algorithm — 1958

Backprop — 1986

Support Vector Machine (SVM) — 1992

ConvNet — 1998

Gradient Boosting Machine (GBM) — 1999

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

Based on personal view.
Source: Wikipedia

2000 – 2010: Arrival of Big Data

flickr

MTurk

kaggle

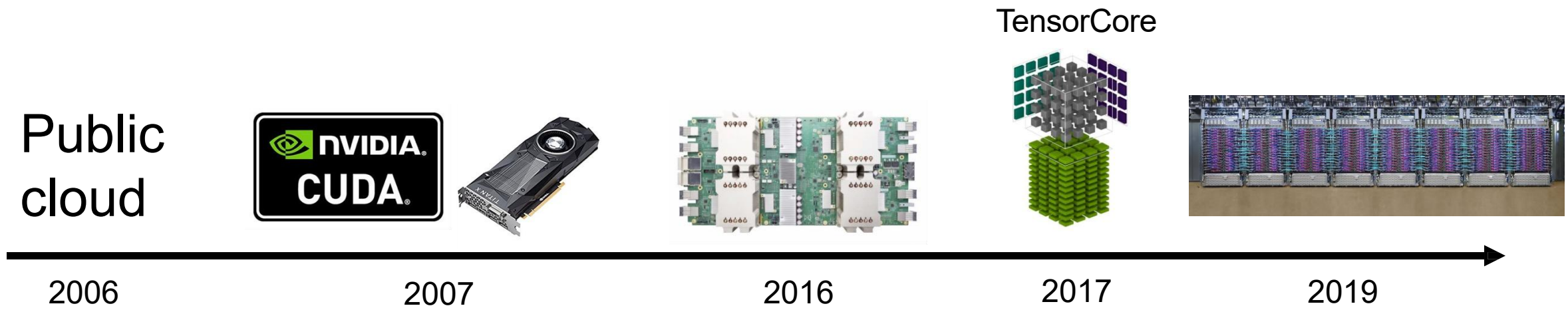
IMAGENET

2001 2004 2005 2009 2010

Data serves as fuel for machine learning models

Based on personal view.
Source: Wikipedia

2006 – Now: Compute and Scaling



Compute scaling

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

Three Pillars of ML Applications

SVM ConvNet

Backprop GBM

ML Research

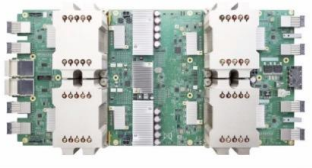
1958

IMAGENET

Data

2000

Public cloud



Compute

2007

AlexNet

Year 2012

ML Research

SGD
Dropout
ConvNet
Pooling

Data

1M labeled images

Compute

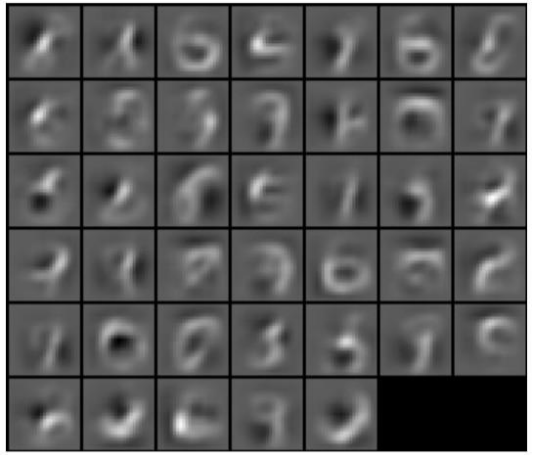
Two Nvidia GTX 580 GPUs

Six days

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

Tianqi Chen's First Deep Learning project

Year 2010



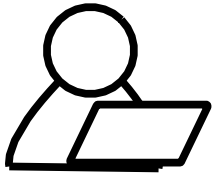
Language	files	blank	comment	code
C	3	84	721	22755
C/C++ Header	43	1773	2616	12324
CUDA	21	1264	1042	7871
C++	17	268	343	1472
MATLAB	9	49	9	245
make	3	26	10	84
Python	2	12	0	42
SUM:	98	3476	4741	44793

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

....

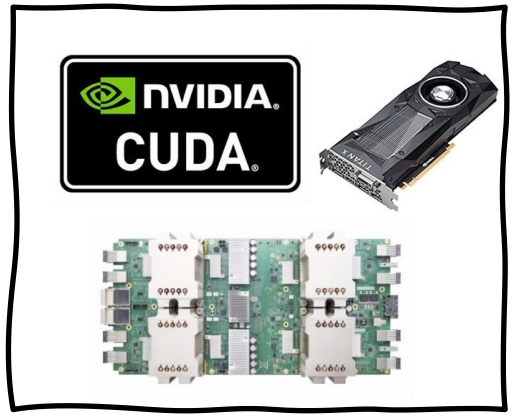
ML Research

44k lines of code

Six months

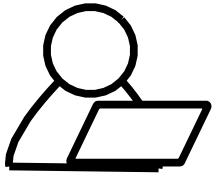
IMAGENET

Data



Compute

Machine Learning Systems



Researcher

ResNet

Transformer

....

ML Research

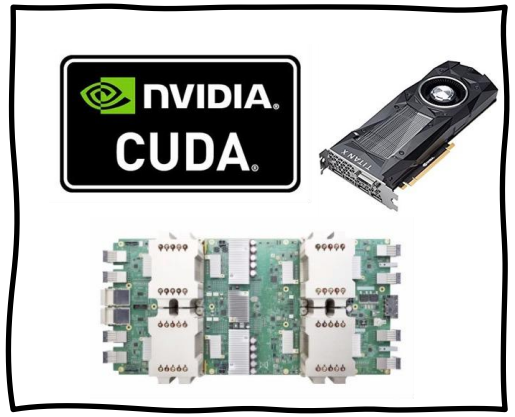
100 lines of python

A few hours

System Abstractions

Systems (ML Frameworks)

Data



Compute

Machine Learning Systems

Researcher

ResNet ….

Transformer

Model

100 lines of python

A few hours

System Abstra

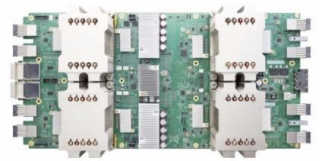
Systems (ML Frameworks)

ML Systems

IMAGENET

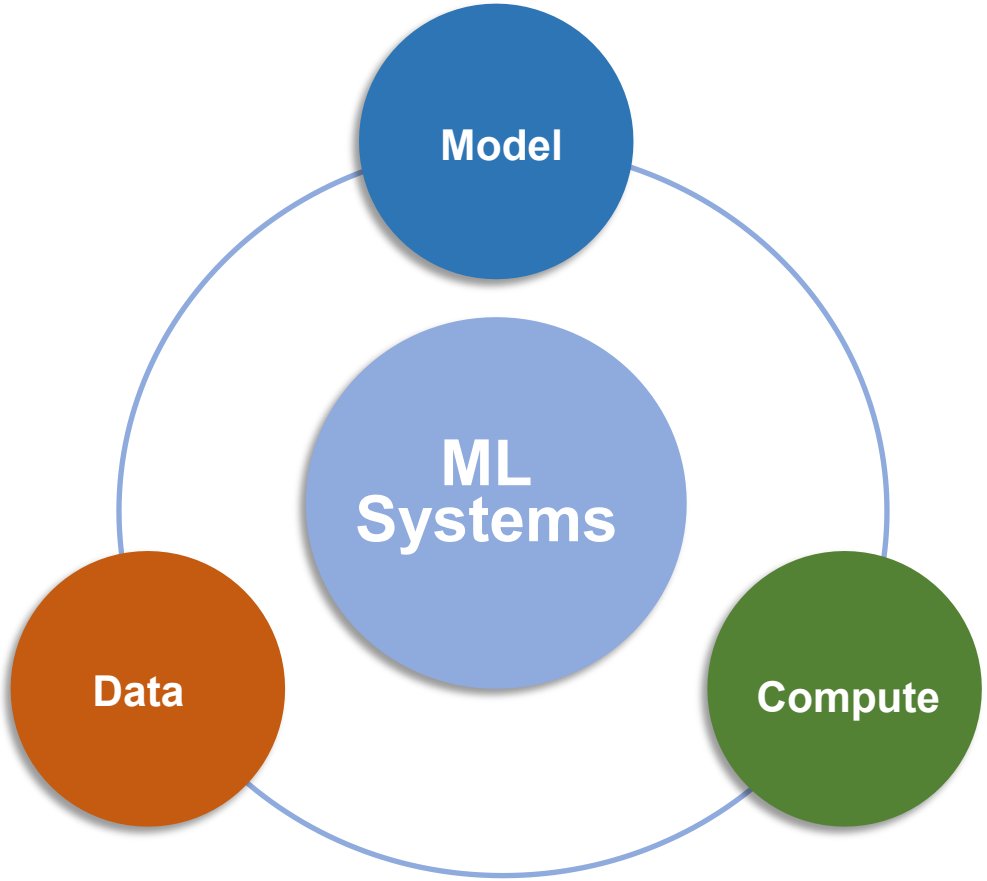
Data

NVIDIA CUDA



Compute

MLSys as a Research Field



Problems / applications

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

A practical problem

To improve pedestrian detection to be X-percent accurate, at Y-ms latency budget with Z-watt hardware

A Typical ML Approach

To improve pedestrian detection to be X-percent accurate, at Y-ms latency budget with Z-watt hardware

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

A Typical Systems Approach

To improve pedestrian detection to be X-percent accurate, at Y-ms latency budget with Z-watt hardware

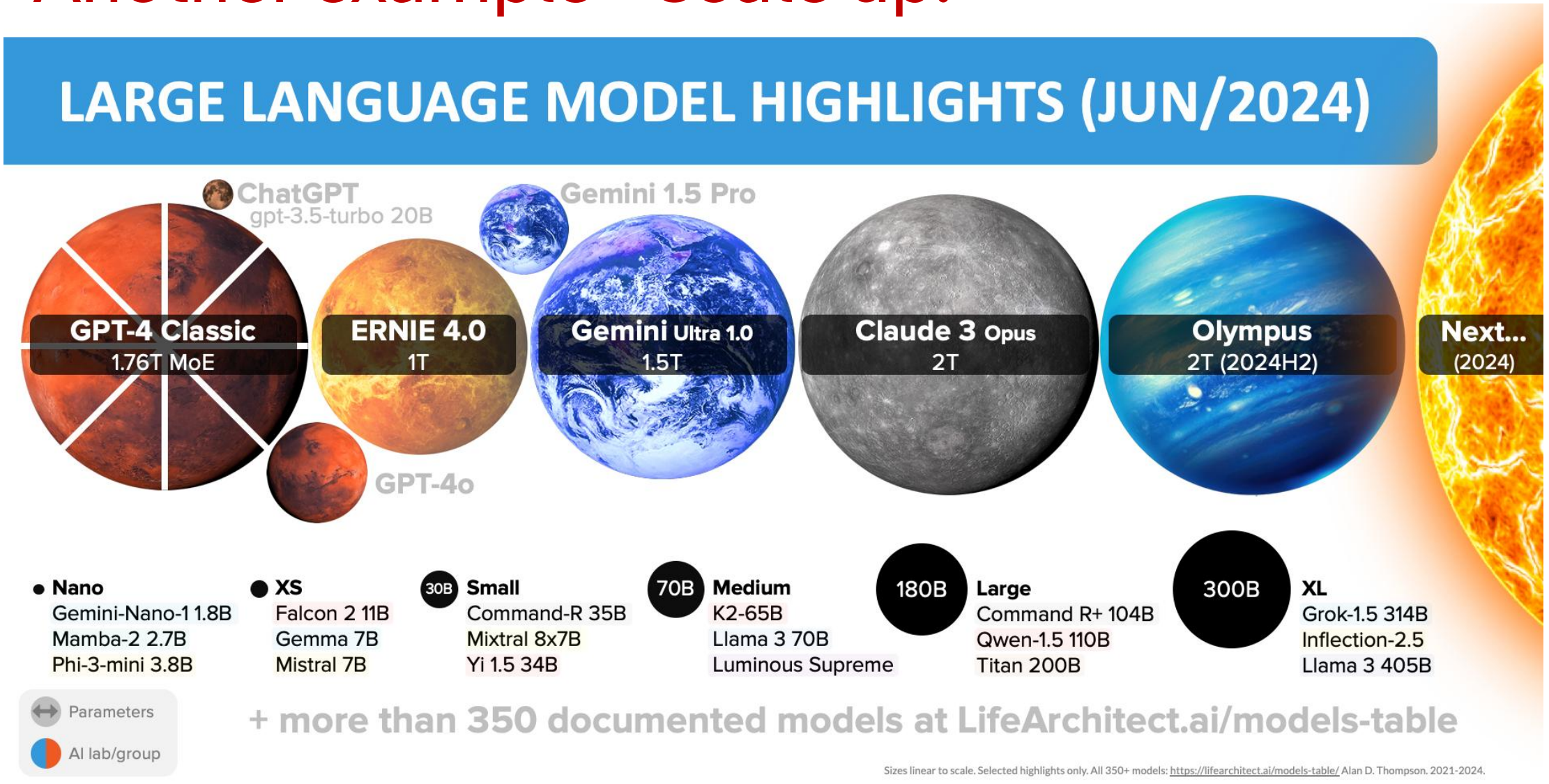
Build a better inference engine to reduce the latency and run more accurate models.

An MLSys Approach

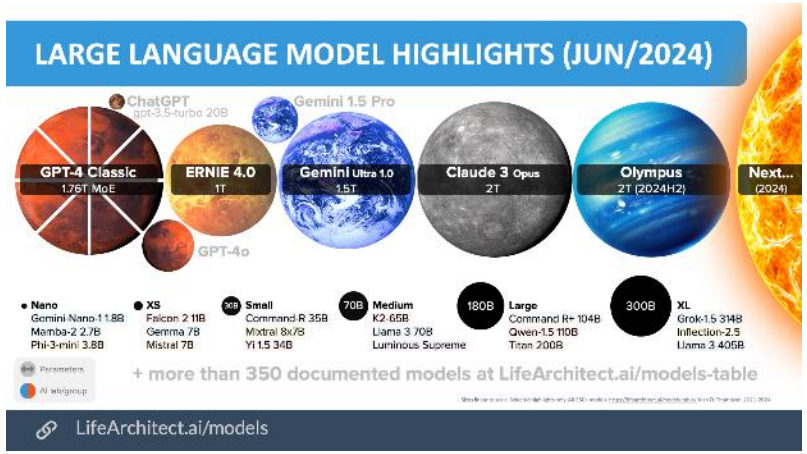
To improve pedestrian detection to be X-percent accurate, at Y-ms latency budget with Z-watt hardware

- **Data**: acquire more sensor data and preprocess them
- **Model**: Develop models that fit the accuracy & latency budget
- **Compute**: Build end-to-end systems for the specific hardware
  - Edge devices & accelerators, sensor data pipelines, decision making

Another example - scale up!



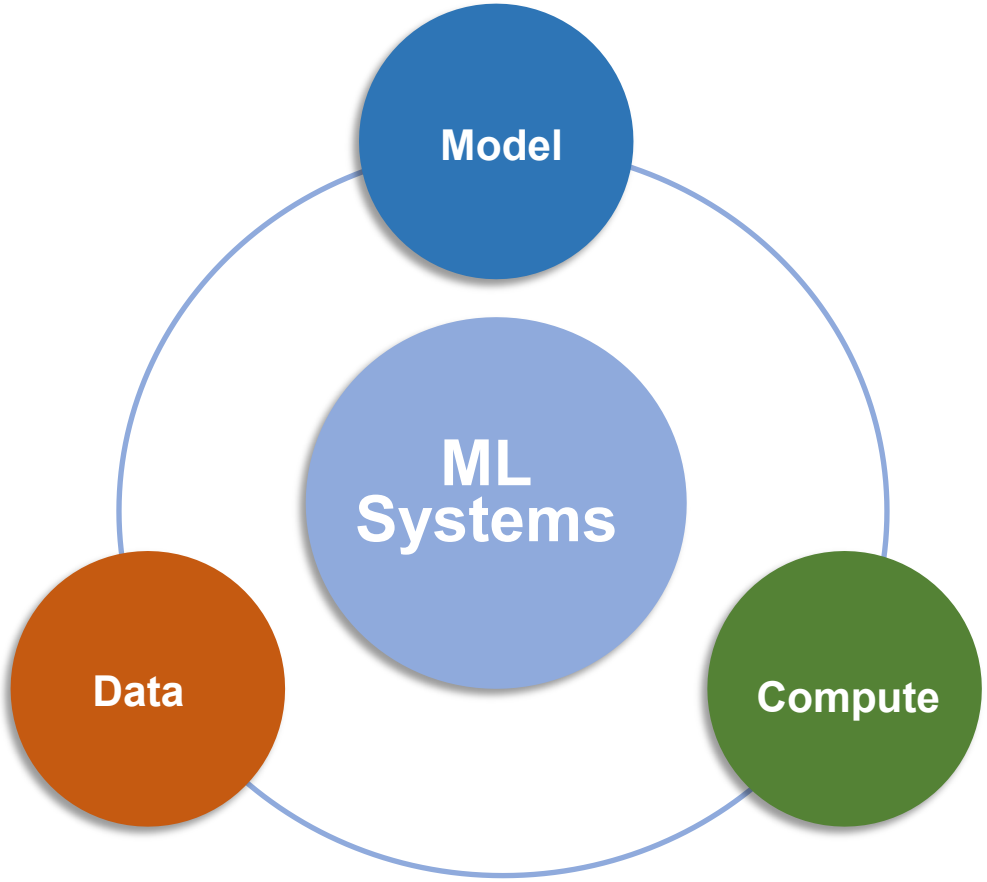
An MLSys Approach



Train an LLM with 1T parameters and maximize model quality

- **Data**: acquire more data and preprocess them
- **Model**: Design models that optimize for the specific model size
- **Systems**: Build end-to-end systems that enable training on a distributed cluster
  - Networking, storage, scheduling, failure recovery etc.

MLSys as an Emerging Research Field



AI Systems Workshop at NeurIPS

MLSys tracks at Systems/DB/Networking conferences

Conference on Machine Learning and Systems (MLSys.org)

MLSys as a Startup Arena

anyscale

LangChain

mosaicML

together.ai

Weaviate

Hugging Face

Lambda

LlamaIndex

Pinecone

Why Study Machine Learning and Systems?

Reason #1 AI is the future. Systems for AI is the foundation.

Reason #2 A full-stack and holistic approach to push the frontier of AI research and production.

Reason #3 Industry: high demand, low supply → high $$$

Outline

- Why machine learning systems

- Some recent topics in ML systems research & production

- Logistics

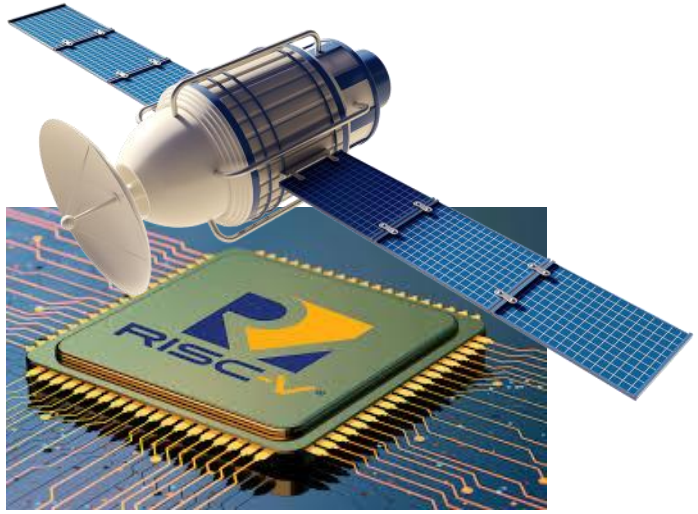
New hardware

- Edge / personal AI devices
- “New” architecture:

  CPU+GPU unified memory
- Problems to solve
  - Improve accuracy / performance & reduce costs
  - Software-hardware co-design

Nvidia DGX Spark

AMD Ryzen AI Max+ 395 CPU+GPU



New AI paradigms

- LLM Pre-training → Post-training
- Modular, (can be) distributed, focusing on end goals

Data infrastructures

Data management for AI :

acquisition, cleaning, structurization,

transformation, annotation, visualization

AI-aided data management & curation

MongoDB.

VectorDB

A PYTHON VECTOR DATABASE YOU JUST NEED - NO MORE, NO LESS.

Azure Data Lake

Data systems for AI :

embedding, storage, indexing,

retrieval, query processing,

LLM-based data analytics

A practical case: AI + finance

- Analyzing 100TB multi-modal data:
  - Time series: stock prices, economic data
  - Unstructured text: news, Twitter, SEC filings in HTMLs
  - Structure info: fundamentals

- LLM tasks:
  - Data preparation
  - Prediction, sentiment analyses, QA
  - Planning, reasoning, simulation

Relationships Between CAP and Financial Performance

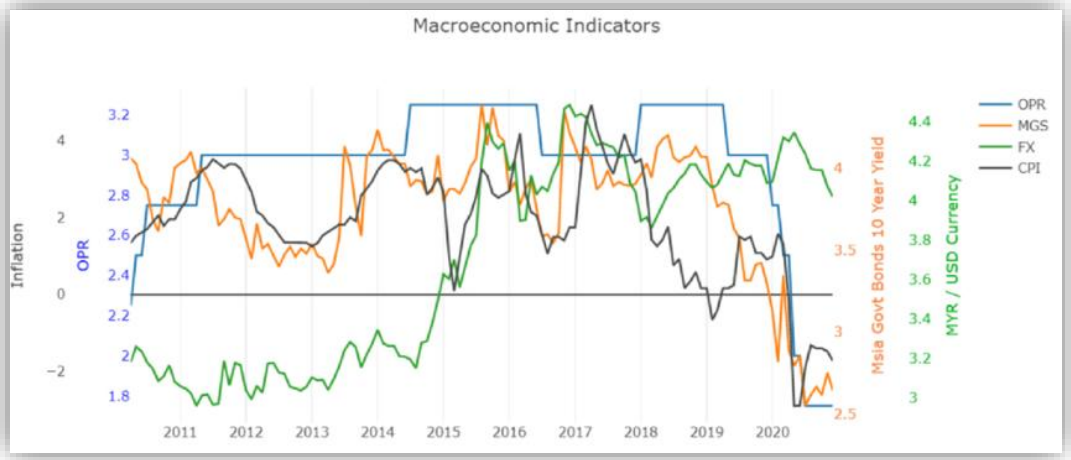
The following graphs illustrate how CAP for our NEOs aligns with the Company's financial performance measures as detailed in the Pay Versus Performance table above for each of Fiscal 2021, 2022, 2023, and 2024, as well as between the TSRs of NVIDIA and the Nasdaq100 Index, reflecting the value of a fixed $100 investment beginning with the market close on January 24, 2020, the last trading day before our Fiscal 2021, through and including the end of the respective listed fiscal years.

NEO CAP versus TSR

NEO CAP versus Net Income & Non-GAAP Operating Income

All information provided above under the "Pay Versus Performance" heading will not be deemed to be incorporated by reference into any filing of the Company under the Securities Act of 1933, as amended, or the Securities Exchange Act of 1934, as amended, whether made before or after the date hereof and irrespective of any general incorporation language in any such filing, except to the extent the Company specifically incorporates such information by reference.

63



Advanced systems

Chef
(LLM)

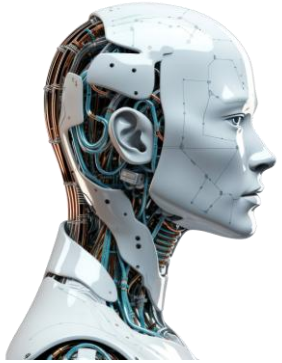
Restaurant
(serving systems)

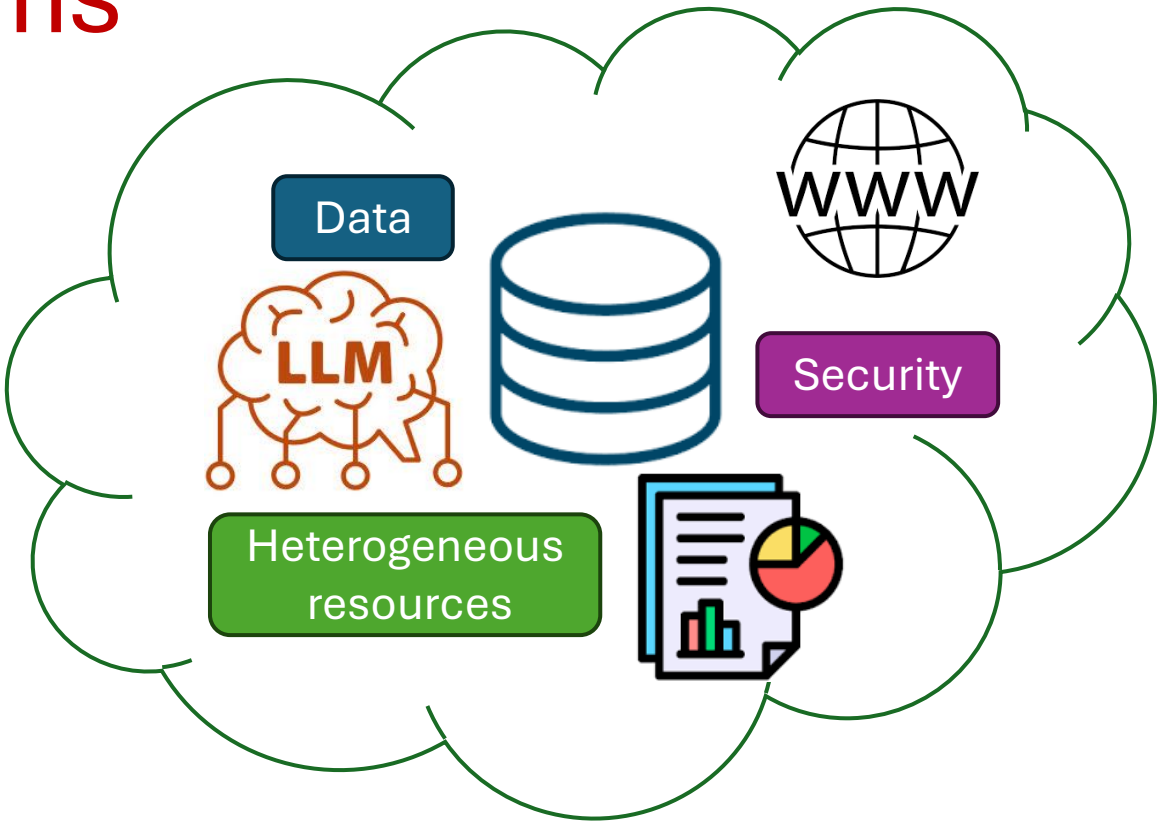
Disney world
(cloud systems)

From serving to cloud systems:

- **Multi-tenancy**: from scaling-up to scaling-out (models, users, applications, tasks etc.)
- **Operations of** large-scale, heterogeneous infrastructures

AI / ML for systems



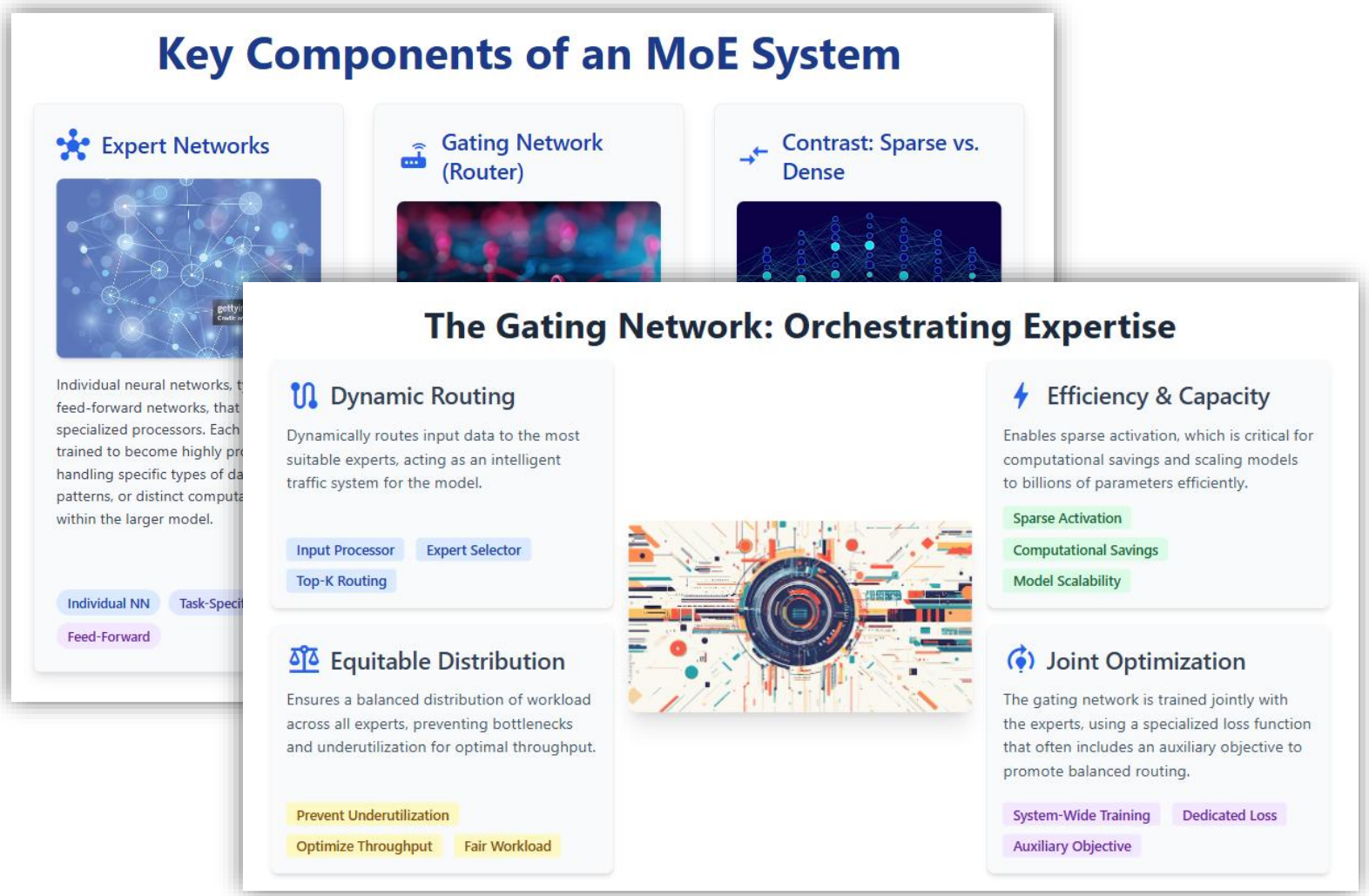


- Problems to solve
  - Use AI to monitor & operate the cloud
  - Use AI/ML to improve individual cloud components
  - Reduce operating costs

Applications

- RAGs
- LLM agents
- Deep research

“Make slides for MoE models” =>



Outline

- Why machine learning systems
- Some recent topics in ML systems research & production
- Logistics

Pre-requisitions

- UG machine learning or equivalent
- UG operating systems or equivalent
- Strong Python programming
- (Optional) C/C++/Rust programming
- This is a system-focused course, not intended for only LLM algorithms / modeling

Course schedule (subject to change)

Week	Date	Lecture	HW schedule	HW Topic
1	08-13	Intro	HW1 out	ML and systems basics
2	08-20	ML sys foundations		
3	08-27	AI framework and autograd	HW1 due (more time)	
4	09-03	Hardware acceleration	HW2 out	AI framework + autograd
5	09-10	Training technologies		
6	09-17	Transformers, attention and optimizations	HW2 due	
	09-24	Recess		
7	10-01	Serving LLMs	HW3 out	LLM inference
8	10-08	Post-training techniques		
9	10-15	Multi-modal models	HW3 due	
10	10-22	Application systems	HW4 out	LLM serving
11	10-29	LLM safety (TBD)		
12	11-05	Cloud systems for AI	HW4 due	
13	11-12	Project presentation		

Assignments and grading

- **Paper reading and discussion**
  - Mandatory, each weak 20%

- **Coding/Written assignments & course projects**
  - HW1 mandatory 20%
  - HW2-4 can be substituted partly or entirely by course projects 60% combined
    e.g.: all HW2-4 and no project , all project, no HW2-4, HW1 + project

- **Course projects** (normalized to 100%)
  - Group of 2-3 people
  - The fewer HW1-3 you take / the more people, the higher expectation
  - Choice & proposal by Week 3. (10%)
  - Mid-term report by the end of Recess week. (20%)
  - Final report by the end of Week 13. (40%)
  - Presentations in Week 13. (30%)
  - Topics: ML **systems** related. Pure ML/AI/CV/NLP projects are not acceptable.

- **Resources**
  - HW0: no GPU is needed. HW1-3 GPU programming as bonus
  - GPU clusters at SOC

Example projects

- Accelerating deep research workflows
- Self-evolving LLM agents
- LLM distillation and alignment
- LLM on personal AI devices

Communications

- Instructor email: luyao@comp.nus.edu.sg
- TA email: j1shen@comp.nus.edu.sg
  noppanat@comp.nus.edu.sg

- Project discussion by appointment

- **Canvas**
  - Notifications
  - Gradebook
  - Homework upload

Disclaimers

- This is the 2nd offering of this course. There are not many similar offerings around the world.
- Industry & open-source world evolving ultra fast.
- The material and outline will likely adjust throughout the semester.
- There will be bugs in the content or assignments.

Concerns & comments?

pe.app/yaolu1