Introduction

Systems for Machine Learning

Chapter 1

https://www.cse.iitb.ac.in/~mythili/sysml/

The impact of AI/ML

- ChatGPT released to public in Nov 2022, followed by widespread adoption of genAI

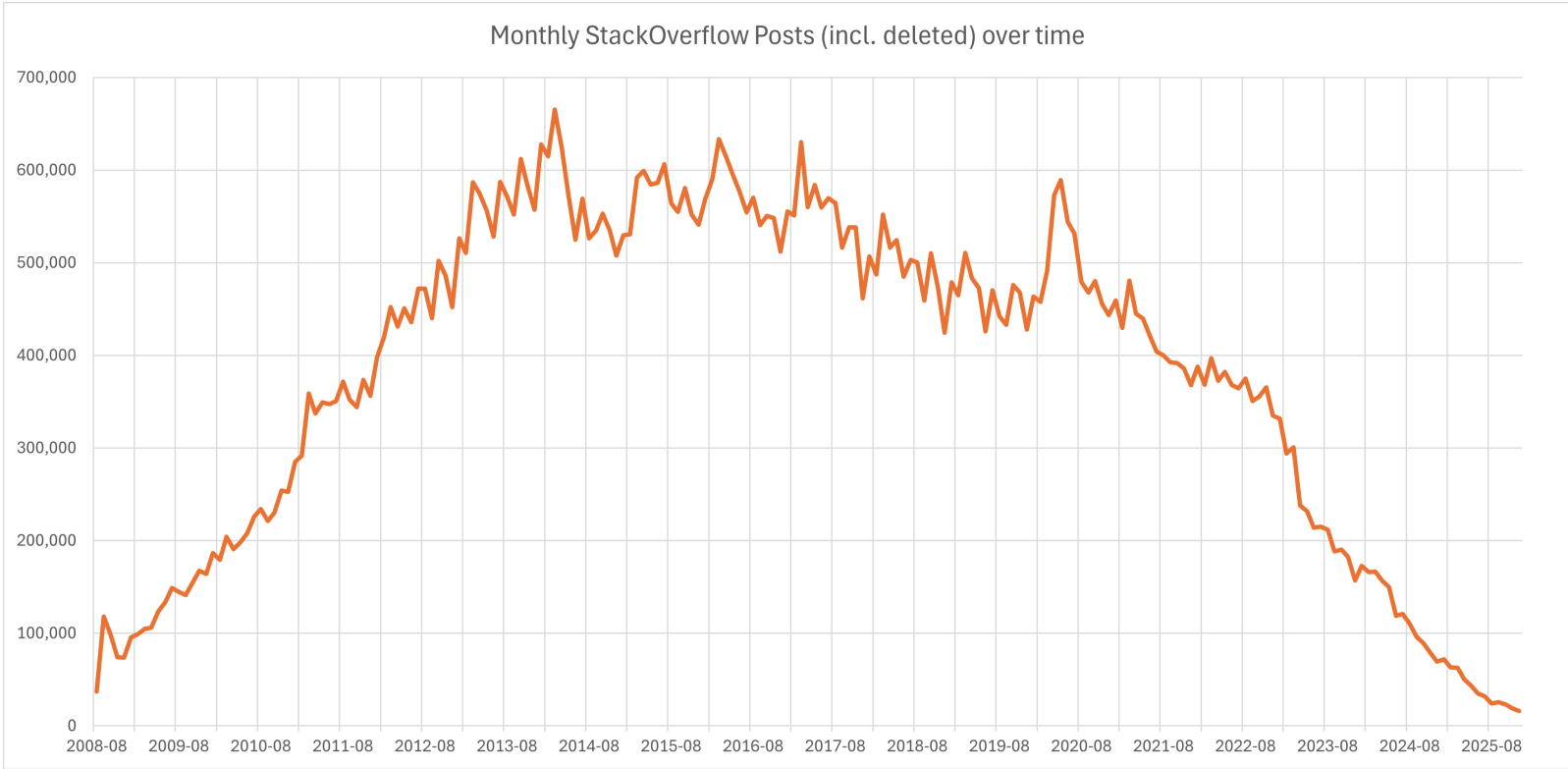
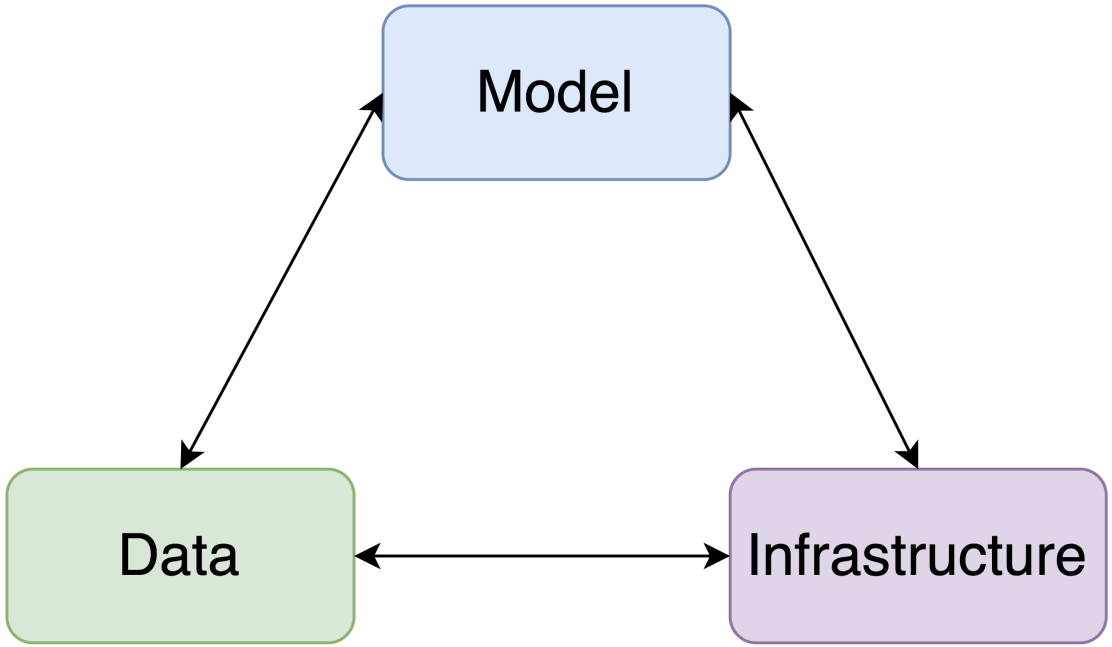


Image Credit: https://meta.stackoverflow.com/questions/437921/how-does-the-continued-decline-in-posts-since-may-25-influence-our-interpretati

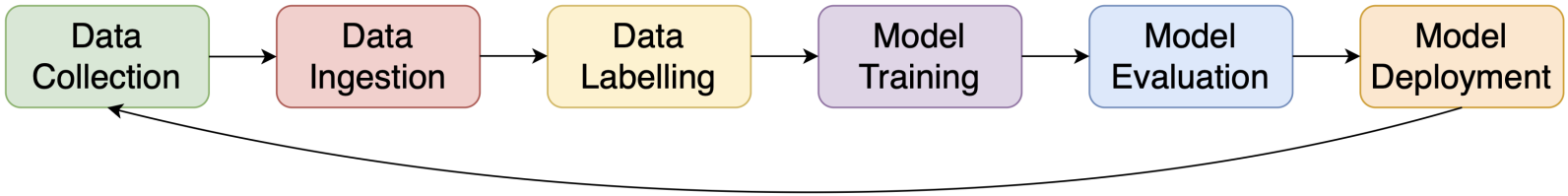
Why Study Systems for Machine Learning?

The Three Pillars of Machine Learning



Machine Learning Lifecycle

- Machine learning lifecycle from data to models
- Every step depends on having scalable hardware and software infrastructure



History of AI and ML

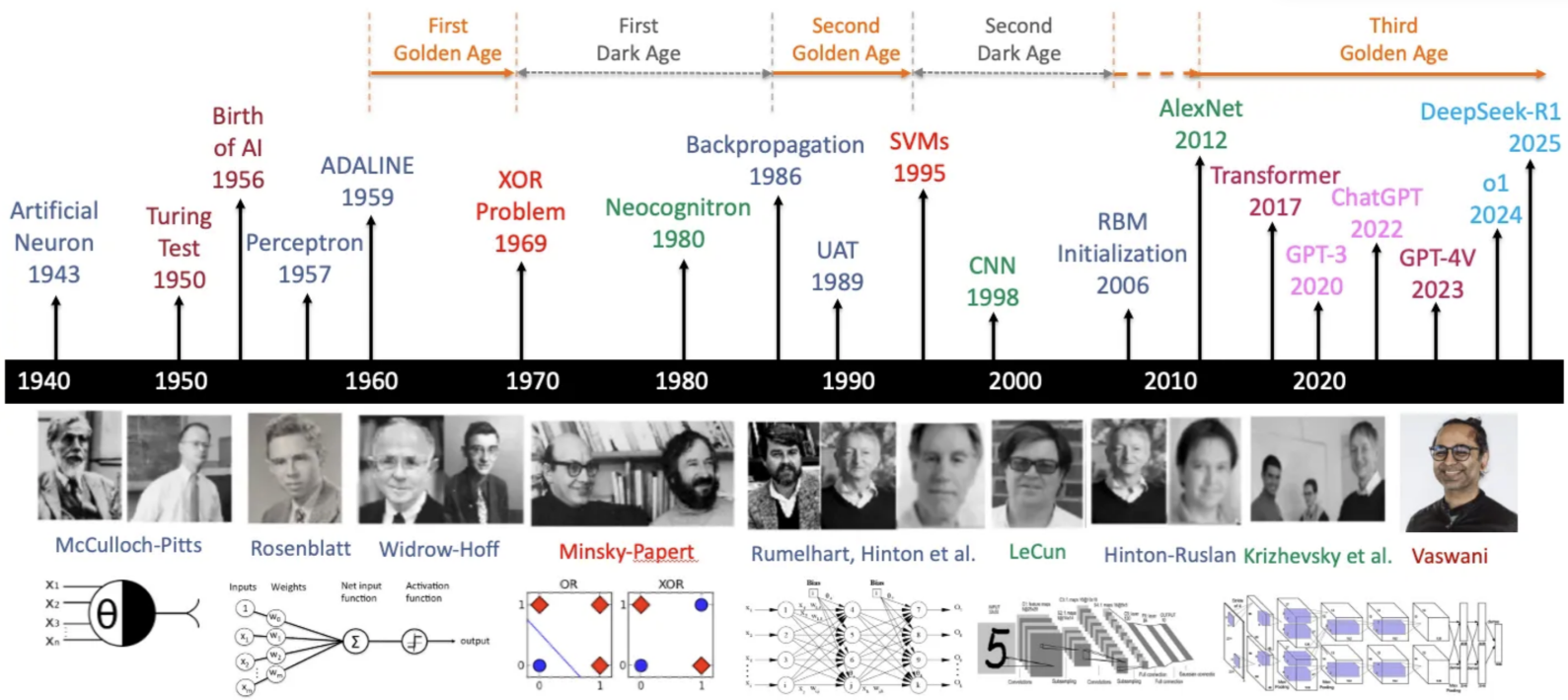


Image Credit: `https://medium.com/@lmpo/a-brief-history-of-ai-with-deep-learning-26f7948bc87b`

History of AI and ML

- Earlier, rule-based expert systems with hand-coded logic
- Later statistical learning systems, e.g., spam filters
- Then to **machine learning**: decision trees, support vector machines, and small neural networks
- **Deep learning**, built on large neural networks, is the newest chapter

The Breakthrough: AlexNet

- Entered into the ImageNet image classification challenge in 2012 by Alex Krizhevsky
- Team trained their network on graphics processing units (**GPUs**) — originally built for rendering video game frames

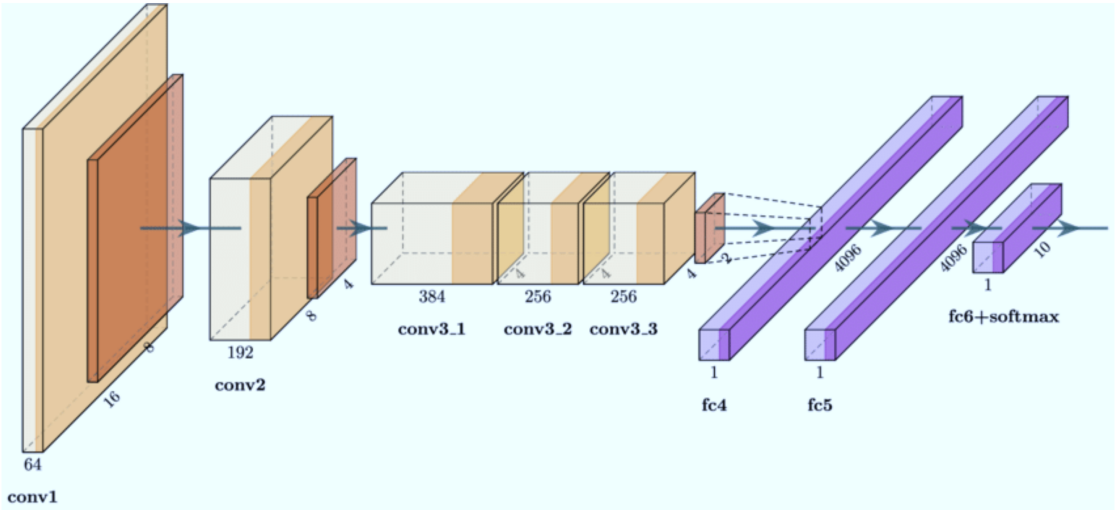


Image Credit: `https://www.kaggle.com/code/yasserh/alexnet`

The Breakthrough: AlexNet

- The core mathematical ideas behind AlexNet were not new in 2012, so what changed?
- **Large datasets** became available through internet and spread of smartphones
- Huge computing power in the form of **GPUs**
- Smaller **algorithmic** breakthroughs made training and inference more stable

Today: ChatGPT, Everyone's Best Friend

- Significant increase in scale, supported by massive infrastructure

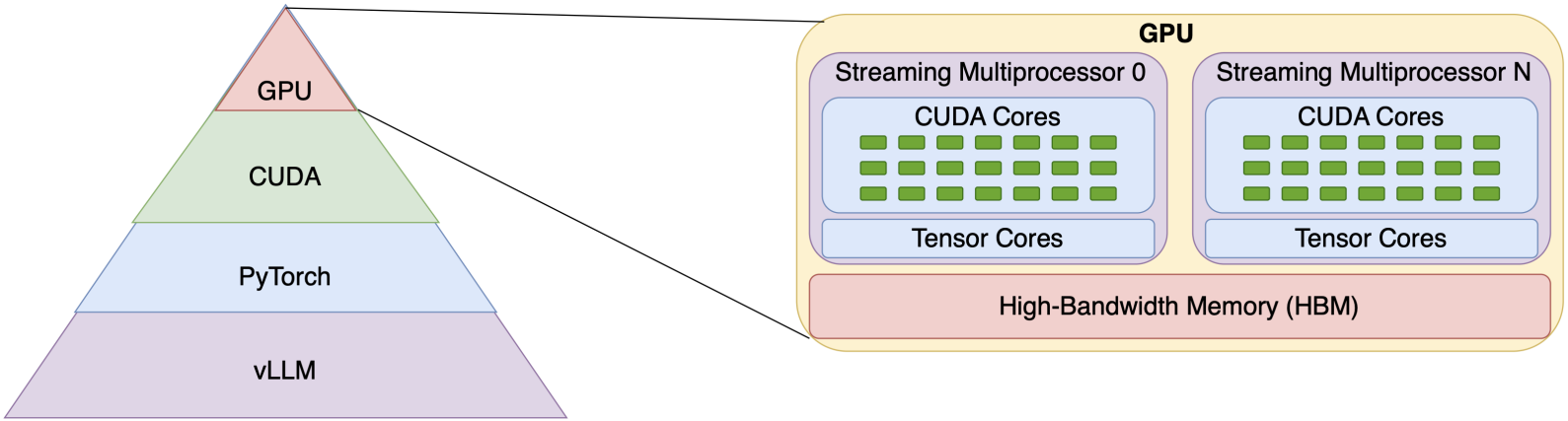
Feature	AlexNet (2012)	GPT-3 (2020)
Model type	Convolutional neural network for image classification	Generative pre-trained transformer for natural language tasks
Parameters	Approximately 60 million	175 billion
Training data size	The ImageNet database, roughly 1.2 million labelled images	Roughly 45 terabytes of raw text, filtered and compressed to about 570 gigabytes
Training hardware	Two Nvidia GTX 580 GPUs, running for about six days	A distributed computing environment using thousands of GPUs running for several weeks

So, why study Systems for ML?

- Modern advances in deep learning depend on underlying scalable systems infrastructure
- System architecture (e.g., hardware memory hierarchy) influences the development of models themselves
- Enormous demand for engineers who sit at the intersection of ML and systems
- Improvement in systems efficiency saves costs and the world's resources!

Software and Hardware Systems

Software and Hardware Systems



GPU Hardware

- CPUs designed to run a huge variety of tasks
- Neural network training and inference need large number of simple, repetitive arithmetic operations
- Better suited to run on GPUs with thousands of small arithmetic cores + specialized **tensor cores** for matmul
- Modern GPUs have high-bandwidth memory (**HBM**) + high-speed inter-GPU connections

Software Stacks

- **Low-level GPU programming**: CUDA
  - Parallel programming C++ extension by **NVIDIA**
  - Ecosystem of optimized CUDA libraries for running ML operations on GPUs
- **High-level programming**: PyTorch, TensorFlow
  - ML models in high-level Python code translate into appropriate CUDA calls
  - Model written as computation graph, automatic differentiation and optimization

Software Stacks

- **Compilers:** Triton, XLA
  - Automatically translate high-level code to low-level optimized GPU instructions
- **Inference engines:** vLLM, SGLang
  - Serving one request at a time wastes an enormous amount of GPU capacity
  - Inference engines efficiently pack many different users' requests into GPU memory via batching

Book Outline

Outline

- **Chapter 1: Introduction**
  - Introduces the three pillars of machine learning and motivates why systems for ML
- **Chapter 2: Overview of Deep Learning Concepts**
  - Major families of neural network architectures
  - Compute and memory footprint analyzed
- **Chapter 3: Programming AI Hardware**
  - GPU compute, memory, networking architecture
  - Introduces CUDA programming framework

Outline

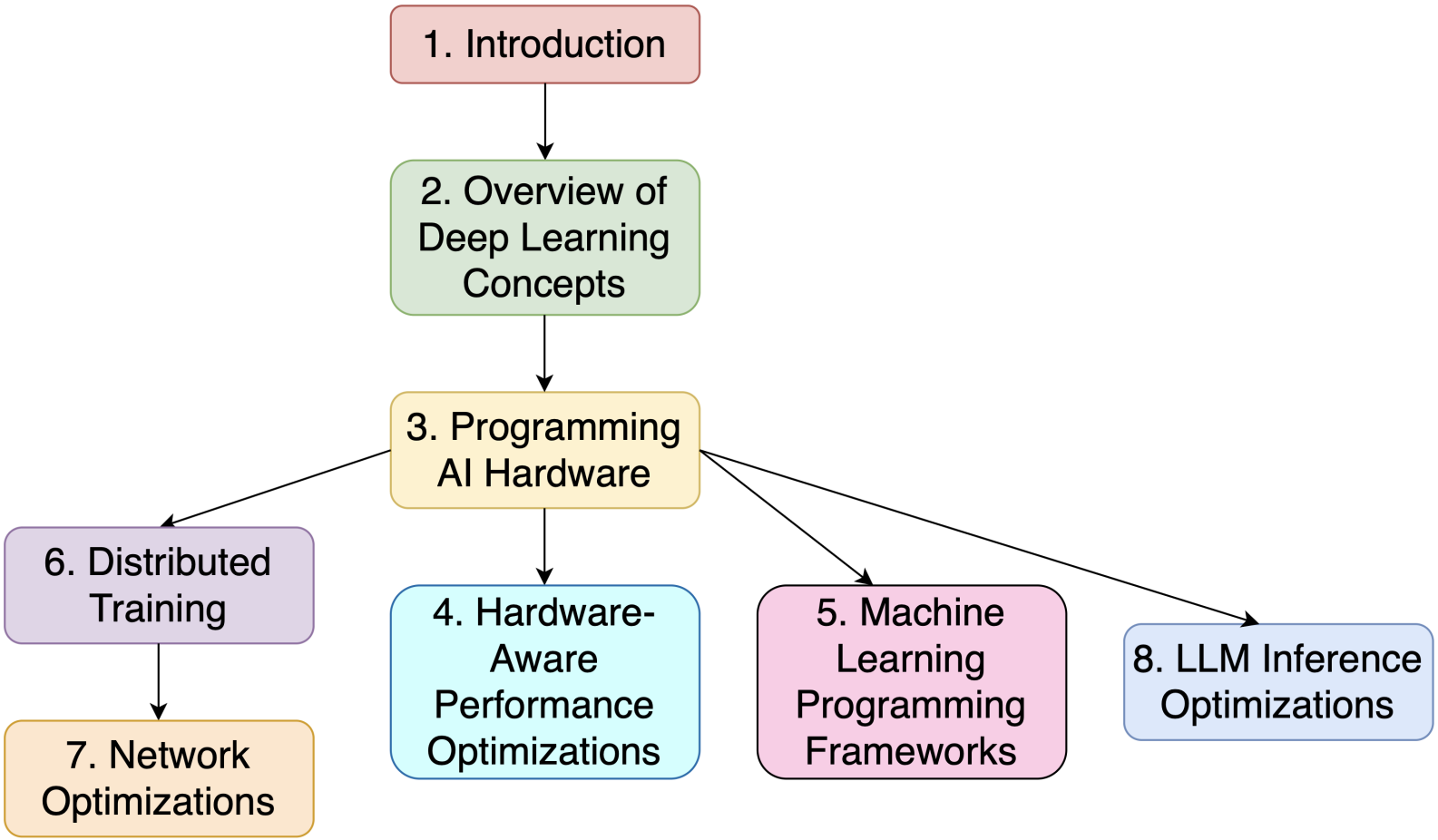
- **Chapter 4: Hardware-Aware Performance Optimizations**
  - Techniques used to speed up ML models on current hardware accelerators
- **Chapter 5: Machine Learning Programming Frameworks**
  - Key ideas in frameworks like PyTorch and TensorFlow
  - Introduces ML compilation
- **Chapter 6: Distributed Training**
  - Introduces data parallelism and model parallelism
  - More specialized strategies and real-world case studies

Outline

- **Chapter 7: Network Optimizations**
  - Network architecture and communication design for AI/ML workloads
  - RDMA and other networking technologies
- **Chapter 8: LLM Inference Optimizations**
  - Serving an already-trained model to real users
  - Baching and scheduling, memory management, other optimizations for large scale inference

How to Use the Book?

Chapter Organization



Practice Problems and Programming Assignments

- Sections in the book end with **practice problems**
- Attempt them with pen and paper before looking at any solutions
- **Programming assignments** present in a companion **GitHub** repository at `https://github.com/mythilivutukuru/SysMLBook`
- Do not lean on AI coding assistants for solving these assignments

Let's Get Started!