NYU

Lecture 0: Introduction and Course Overview

CSCI-GA 3033

Special Topics: Efficient AI Computing: Algorithm and Implementation

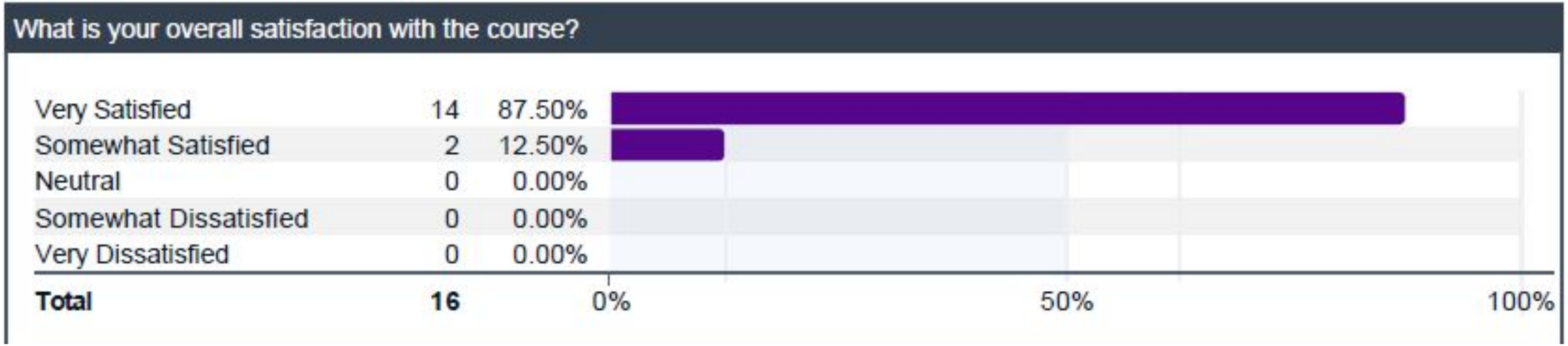
Self Introduction

- Assistant Professor, NYU, ECE & CS, lead **System & AI (SAI) lab**.
- A senior research scientist at Meta, 2022-2024.
- Academic trajectory
  - University of Toronto
    - Bachelor and Master in ECE
    - Master in Statistics
  - Harvard University
    - PhD in CS
- Research Interest:
  - Efficient AI Algorithm
  - AI Hardware Accelerator
  - AR/VR System

Course Information

- Course website: https://www.saiqianzhang.com/COURSE/
- I use Brightspace to post announcements and grades
- I provide an online zoom meeting option for people interested in auditing the class. However, enrolled students are required to attend in person unless special condition.
- A suggested reading list which contains interesting papers can be found here.
- Discussion groups has been created in the Brightspace
- Course email: efficientaiaccelerator@gmail.com

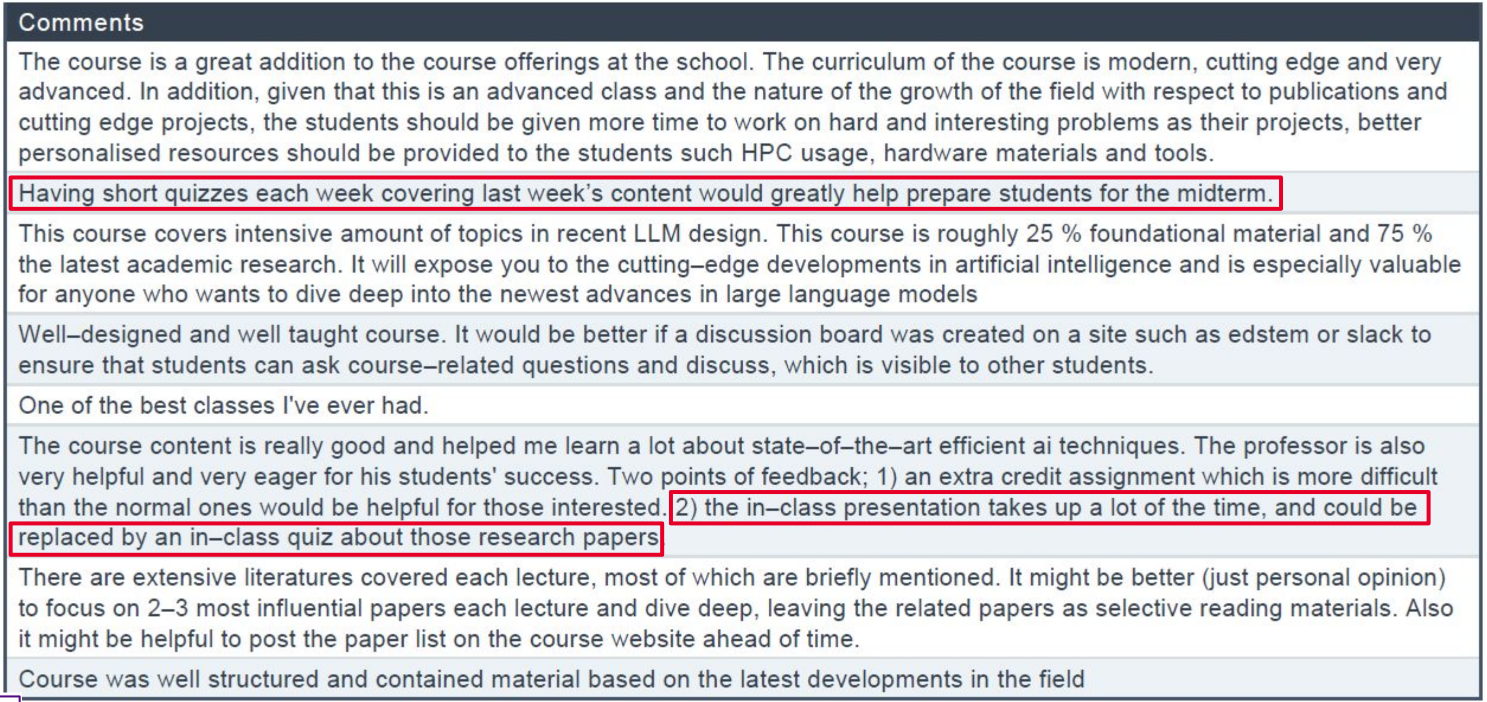
Course Feedback from Spring 2025



What is your overall satisfaction with the course?

Very Satisfied	14	87.50%	
Somewhat Satisfied	2	12.50%	
Neutral	0	0.00%	
Somewhat Dissatisfied	0	0.00%	
Very Dissatisfied	0	0.00%	
Total	**16**		

Course Feedback from Spring 2025



Comments
The course is a great addition to the course offerings at the school. The curriculum of the course is modern, cutting edge and very advanced. In addition, given that this is an advanced class and the nature of the growth of the field with respect to publications and cutting edge projects, the students should be given more time to work on hard and interesting problems as their projects, better personalised resources should be provided to the students such HPC usage, hardware materials and tools.
Having short quizzes each week covering last week's content would greatly help prepare students for the midterm.
This course covers intensive amount of topics in recent LLM design. This course is roughly 25 % foundational material and 75 % the latest academic research. It will expose you to the cutting–edge developments in artificial intelligence and is especially valuable for anyone who wants to dive deep into the newest advances in large language models
Well–designed and well taught course. It would be better if a discussion board was created on a site such as edstem or slack to ensure that students can ask course–related questions and discuss, which is visible to other students.
One of the best classes I've ever had.
The course content is really good and helped me learn a lot about state–of–the–art efficient ai techniques. The professor is also very helpful and very eager for his students' success. Two points of feedback; 1) an extra credit assignment which is more difficult than the normal ones would be helpful for those interested. 2) the in–class presentation takes up a lot of the time, and could be replaced by an in–class quiz about those research papers
There are extensive literatures covered each lecture, most of which are briefly mentioned. It might be better (just personal opinion) to focus on 2–3 most influential papers each lecture and dive deep, leaving the related papers as selective reading materials. Also it might be helpful to post the paper list on the course website ahead of time.
Course was well structured and contained material based on the latest developments in the field

Course Information

- The course will involve 13 lectures, 3 coding assignments, 1 final project, 1 midterm exam and in-class quiz.
  - In-class quiz (10%)
  - Assignments (30%): total three of them, each counts 10%
  - Midterm (30%)
  - Final project (30%)
    - Project Proposal (5%) (1 page)
    - Final Presentation (15%)
    - Final Report (10%)
- Readings:
  - Course notes and papers (optional)
  - (reference) Goodfellow, Ian. "Deep learning." (2016). https://www.deeplearningbook.org/
- Lecture time:
  - Wednesday: 7:10pm-9:10pm
- Office hour:
  - Friday: 1:30pm-2:30pm, or by appointment (Zoom)

Course Assistant/Grader

Shawn Yin (CA)

Office hour: Monday
1:00pm-2:00pm
(Zoom)

Xiwen Min (CA)

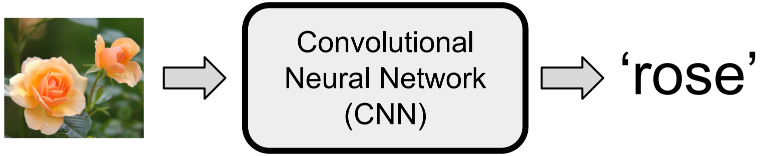
Office hour: Thursday
10:00am-11:00am
(Zoom)

Yunhai Hu (Grader)

NYU SAI LAB

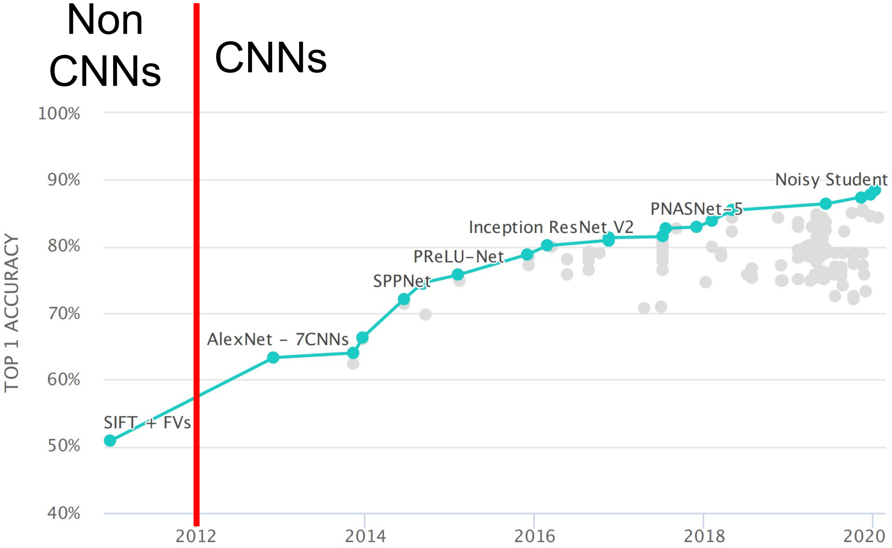
Life is Powered by Deep Learning

- Deep Neural Networks (DNNs) have achieved state-of-the-art performance across a variety of domains
  - ○ Image Recognition
  - ○ Video Processing
  - ○ Natural Language Processing
  - ○ Autonomous Driving


Convolutional Neural Network (CNN)

‘rose’

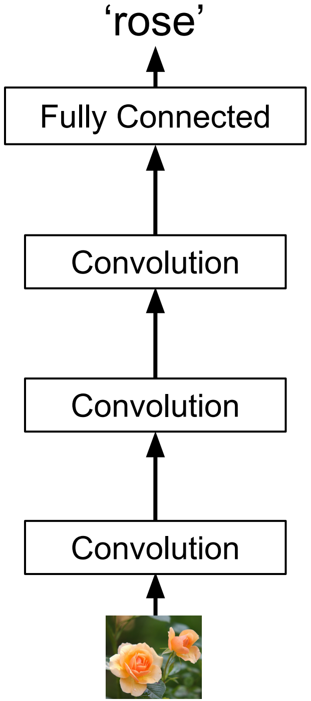
Image Classification



- More desirable modern services are enabled by DNN

How Deep Neural Network is Executed?

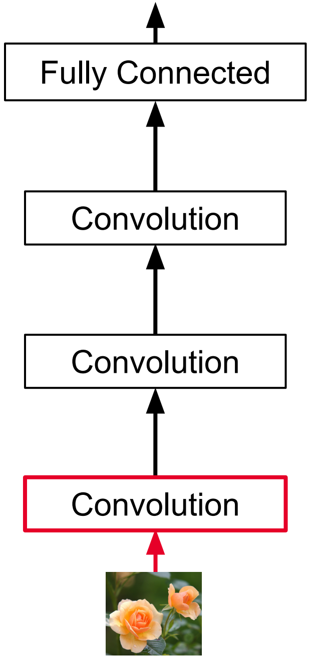
- Use a Convolutional Neural Network (CNN) as an example
- This CNN contains four layers
  - 3 convolutional layers
  - 1 fully connected layer



CNN with 4 layers

NYU SAI LAB

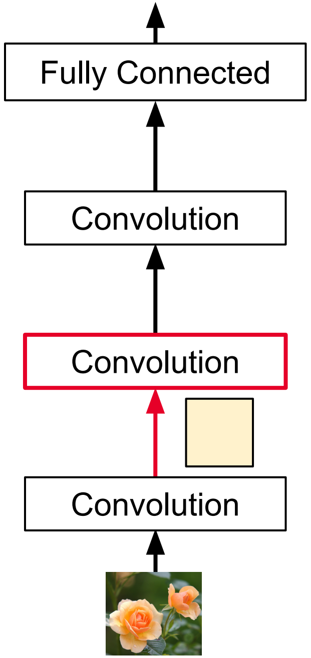
How Deep Neural Network is Executed?



CNN with 4 layers

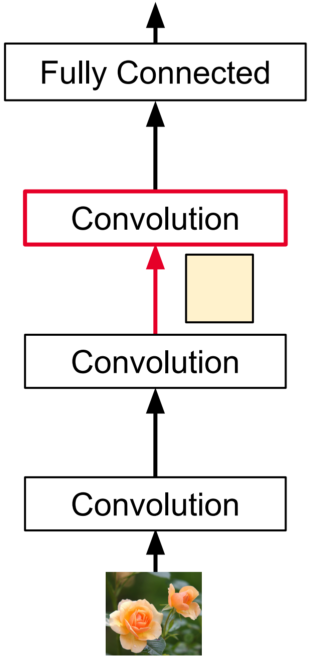
NYU SAI LAB

How Deep Neural Network is Executed?



CNN with 4 layers

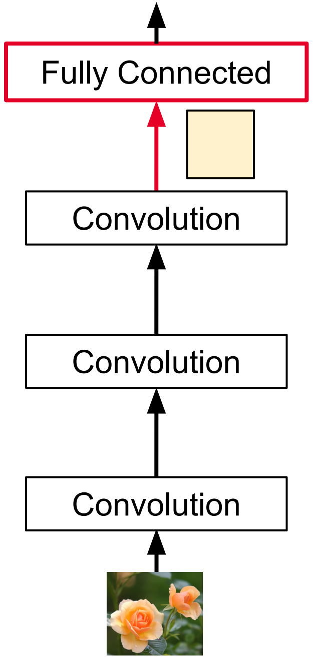
How Deep Neural Network is Executed?



CNN with 4 layers

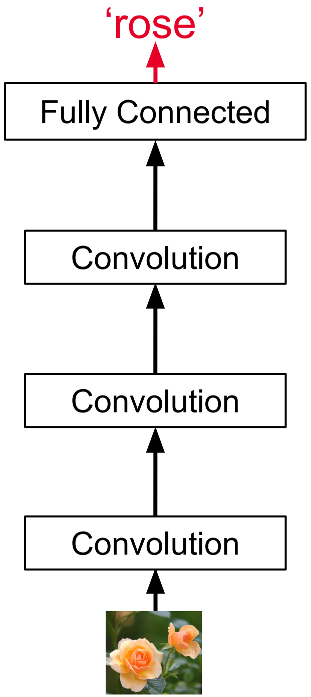
NYU SAI LAB

How Deep Neural Network is Executed?



CNN with 4 layers

How Deep Neural Network is Executed?



‘rose’

Fully Connected

Convolution

Convolution

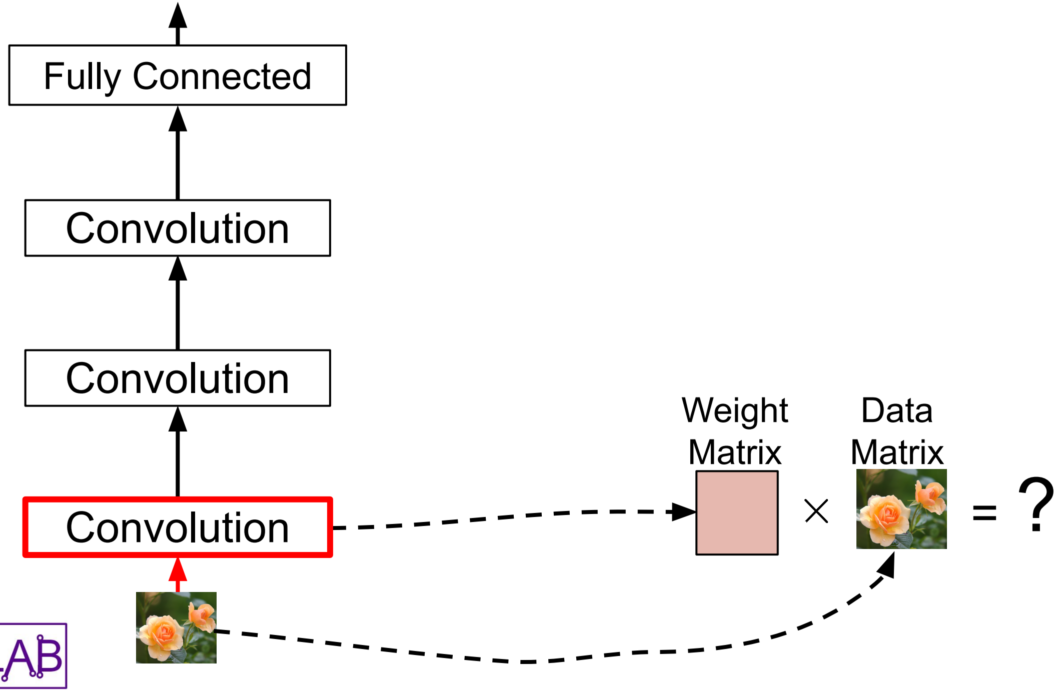
Convolution

CNN with 4 layers

DNN Execution: A Matrix View

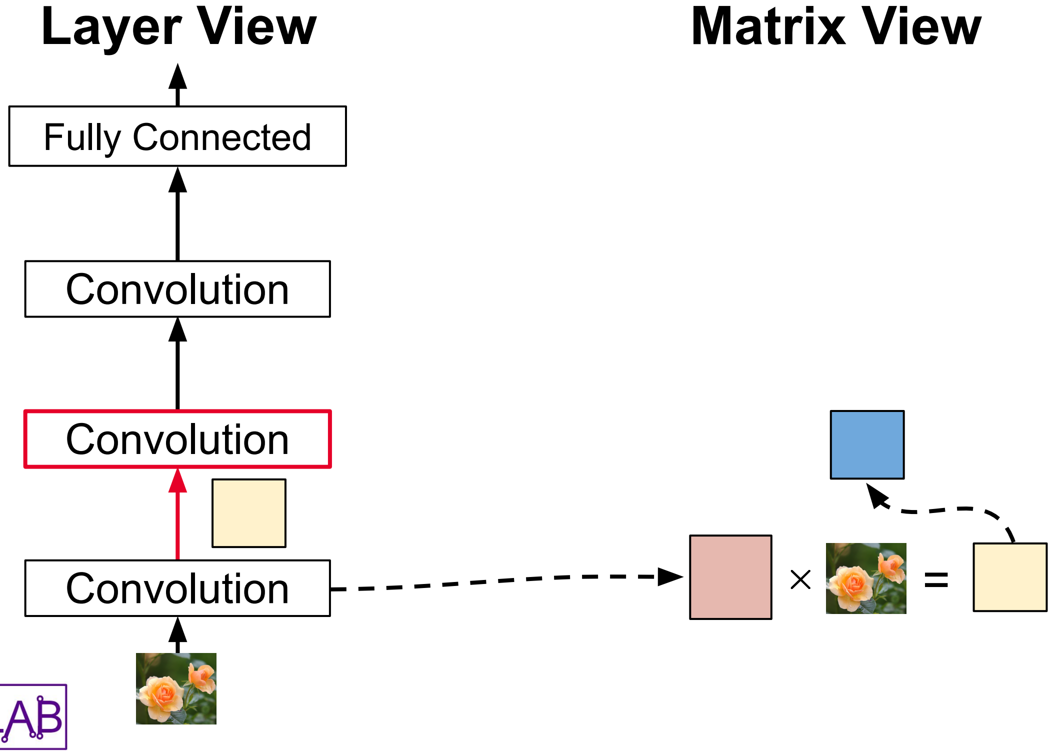
Layer View

Matrix View



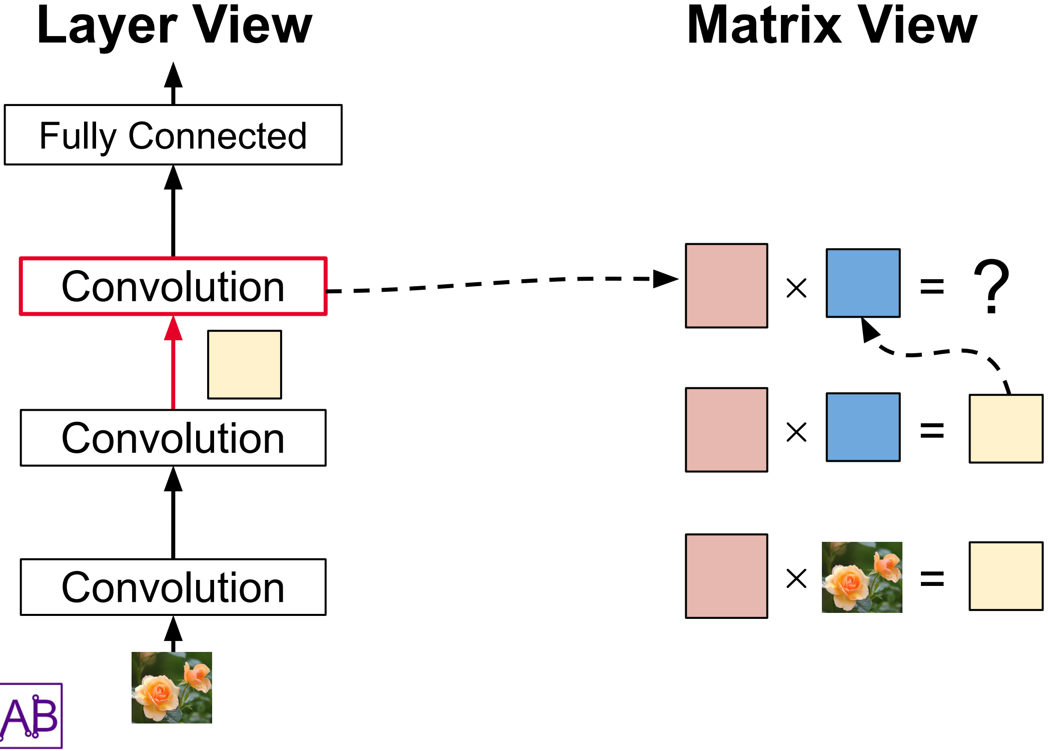
- Weight matrices are **learned** during training

DNN Execution: A Matrix View



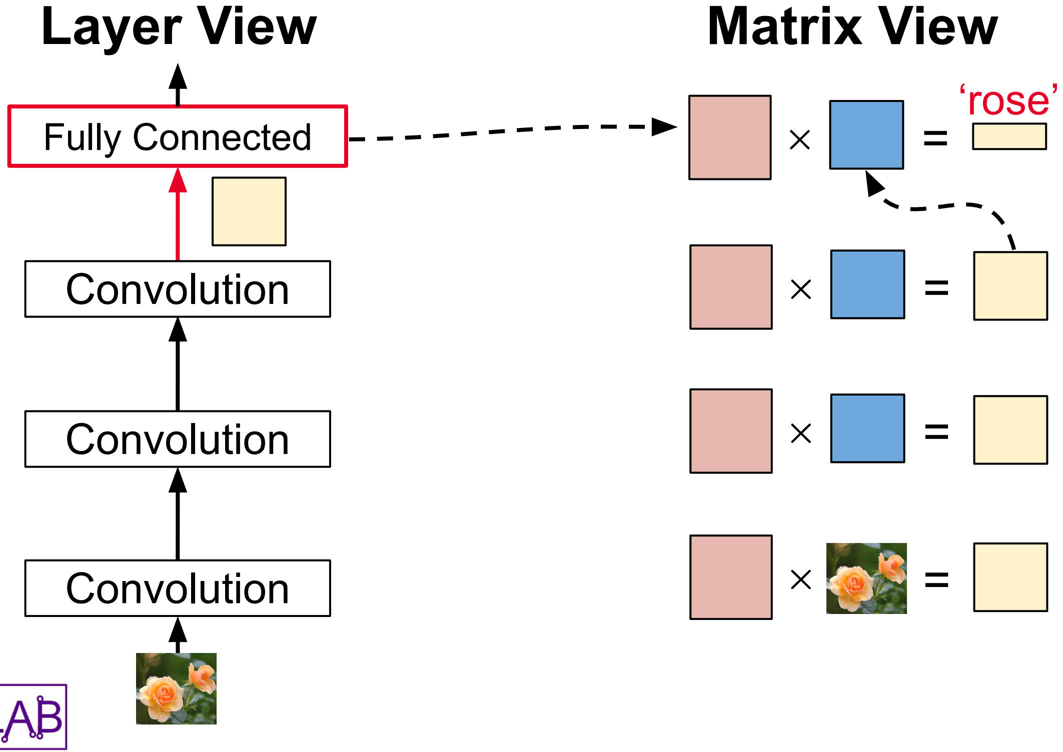
- Weight matrices are **learned** during training

DNN Execution: A Matrix View



- Remaining layers follow this pattern.

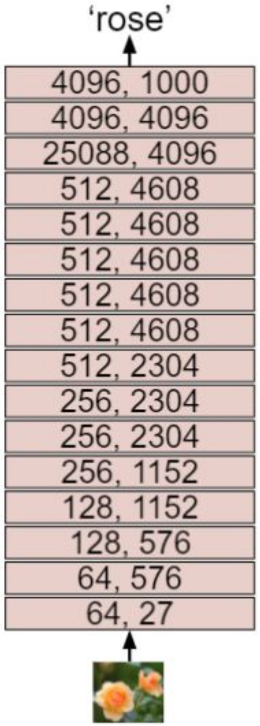
DNN Execution: A Matrix View



- Remaining layers follow this pattern

Deployment of DNN: Problems

- The majority of computation workloads for DNN inference involves a series of **matrix multiplications.**



‘rose’

4096, 1000

4096, 4096

25088, 4096

512, 4608

512, 4608

512, 4608

512, 4608

512, 4608

512, 2304

256, 2304

256, 2304

256, 1152

128, 1152

128, 576

64, 576

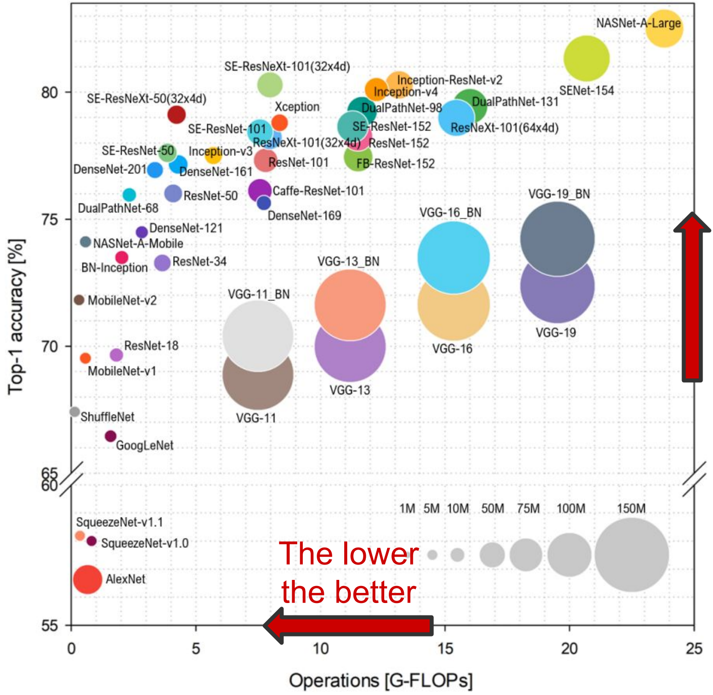
64, 27

VGG-16 is a CNN with over 150M weights across 16 matrices

Deployment of DNN: Problems

- DNN suffers due to:
  - High energy consumption
  - High processing latency
  - High storage cost
- DNN needs to maintain high accuracy

20B multiply/adds per image

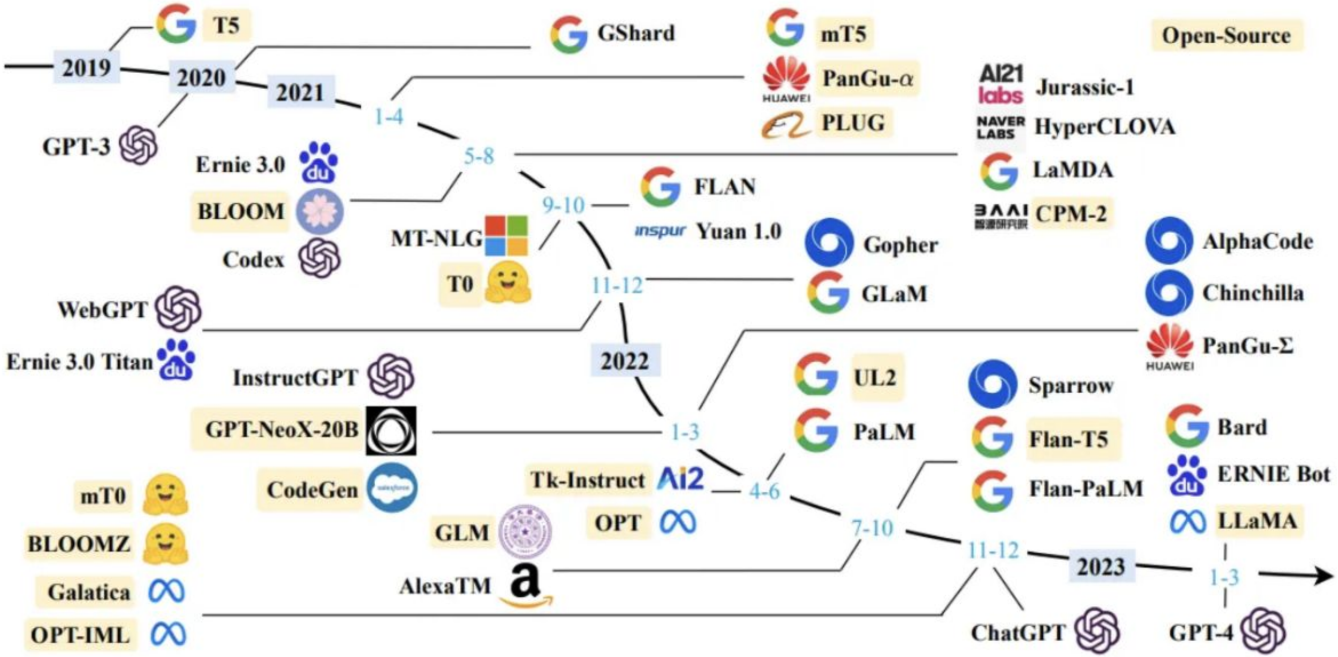


The lower the better

The higher the better

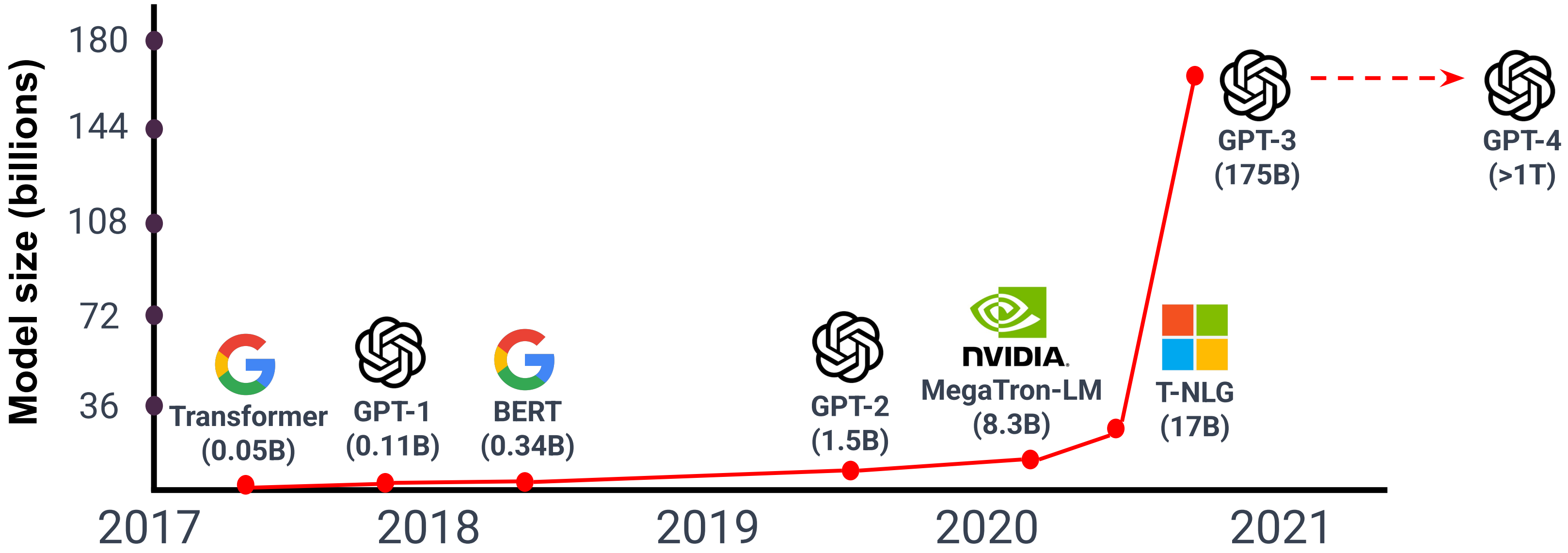
Bianco, Simone, et al. "Benchmark analysis of representative deep neural network architectures." *IEEE Access* 6 (2018): 64270-64277.

The Era of Large Models (LMs)



NYU SAI LAB

Cost of Large Models



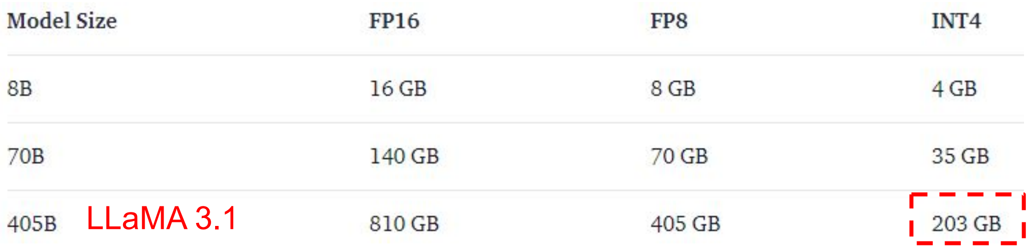
- $1.4e^{12}$ FLOPs to execute GPT-2.

NYU SAI LAB

The Cost of Large Models

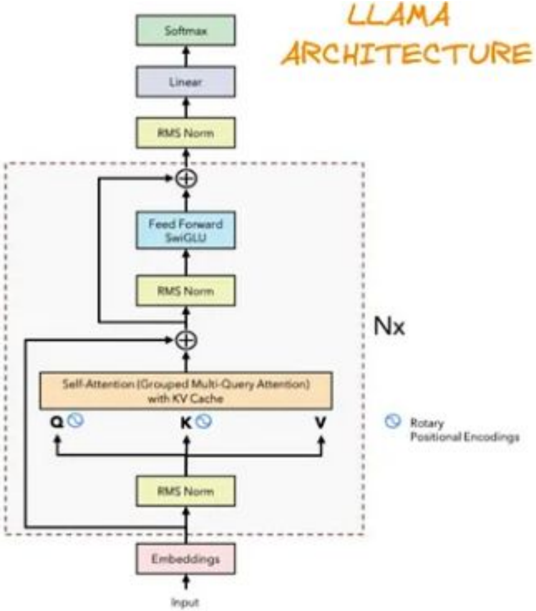
- Training GPT-4 required **25,000 A100 GPUs** over several weeks.
- **Cost**: Renting a single high-end GPU on cloud services like AWS can cost **$3–$5 per hour.** Training GPT-4 is estimated to cost **$63-100 million** on cloud computing resources.

Efficient AI: An Emerging Area



Model Size	FP16	FP8	INT4
8B	16 GB	8 GB	4 GB
70B	140 GB	70 GB	35 GB
405B LLaMA 3.1	810 GB	405 GB	203 GB

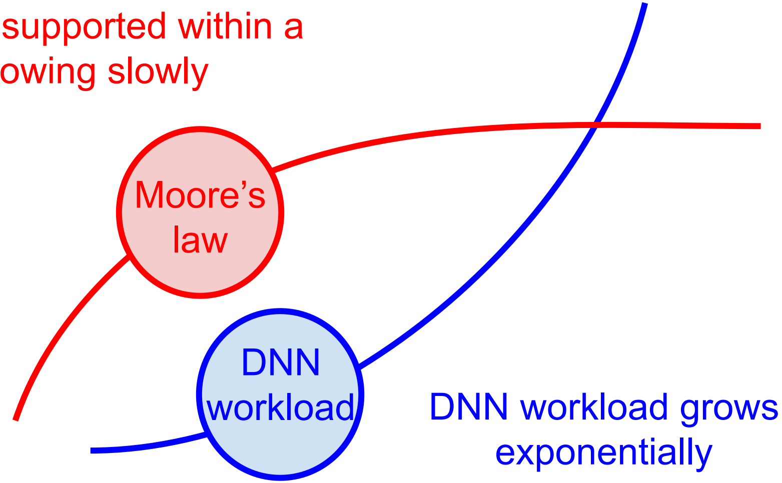
Design more aggressive and efficient AI model is of paramount importance



NYU SAI LAB

Efficient AI: An Emerging Area

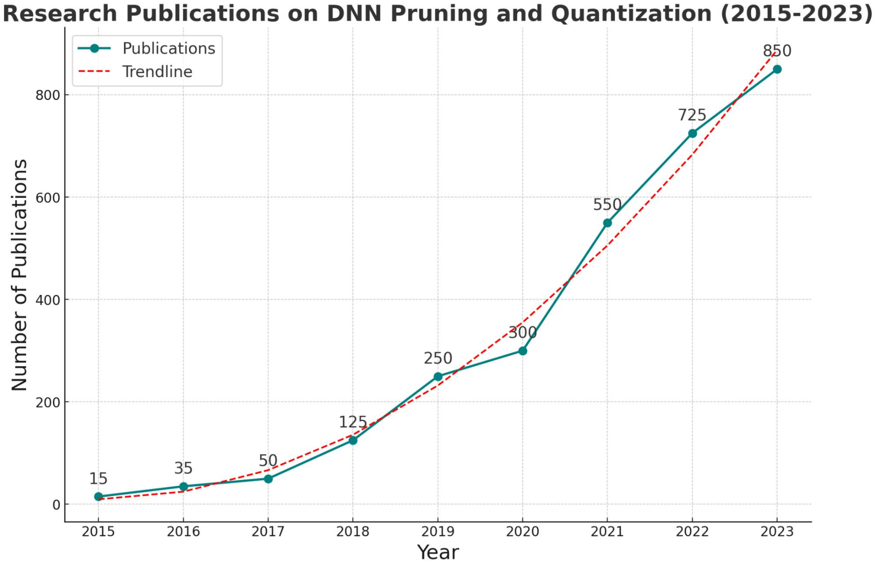
The amount of compute supported within a hardware unit is growing slowly



DNN workload grows exponentially

How to reduce the compute while maintaining a good DNN accuracy?

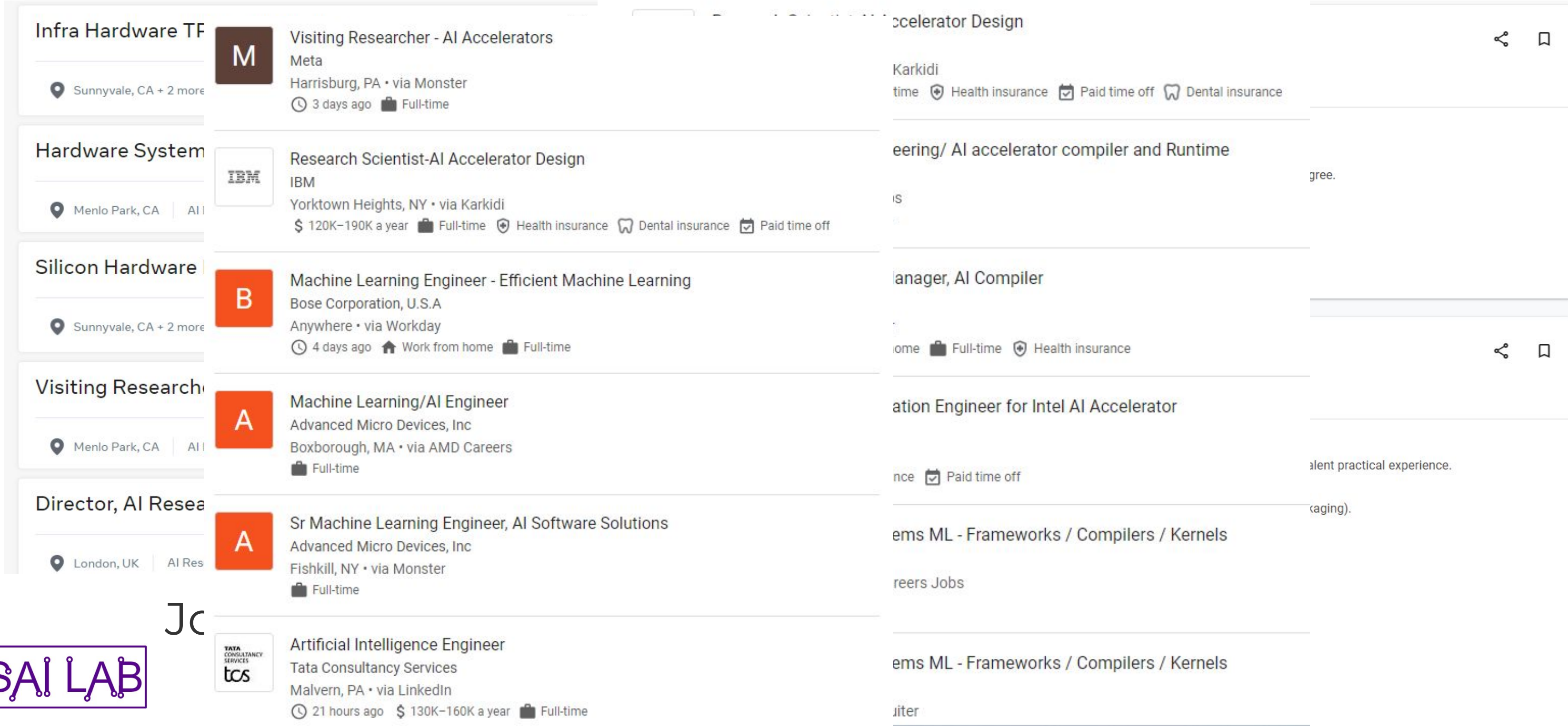
Efficient AI: An Emerging Area



- Efficient AI has become one of the most popular areas in AI community.
- The recent emergence of large models has further heightened the need for efficient AI.

NYU SAI LAB

Efficient AI: An Emerging Area



Infra Hardware TF

Sunnyvale, CA + 2 more

Hardware System

Menlo Park, CA | AI I

Silicon Hardware I

Sunnyvale, CA + 2 more

Visiting Research

Menlo Park, CA | AI I

Director, AI Resea

London, UK | AI Res

Visiting Researcher - AI Accelerators
Meta
Harrisburg, PA • via Monster
3 days ago · Full-time

Research Scientist-AI Accelerator Design
IBM
Yorktown Heights, NY • via Karkidi
$ 120K–190K a year · Full-time · Health insurance · Dental insurance · Paid time off

Machine Learning Engineer - Efficient Machine Learning
Bose Corporation, U.S.A
Anywhere • via Workday
4 days ago · Work from home · Full-time

Machine Learning/AI Engineer
Advanced Micro Devices, Inc
Boxborough, MA • via AMD Careers
Full-time

Sr Machine Learning Engineer, AI Software Solutions
Advanced Micro Devices, Inc
Fishkill, NY • via Monster
Full-time

Artificial Intelligence Engineer
Tata Consultancy Services
Malvern, PA • via LinkedIn
21 hours ago · $ 130K–160K a year · Full-time

ccelerator Design
Karkidi
time · Health insurance · Paid time off · Dental insurance

eering/ AI accelerator compiler and Runtime

ıs

lanager, AI Compiler

ıome · Full-time · Health insurance

ation Engineer for Intel AI Accelerator

nce · Paid time off

ems ML - Frameworks / Compilers / Kernels

reers Jobs

ems ML - Frameworks / Compilers / Kernels

ıiter

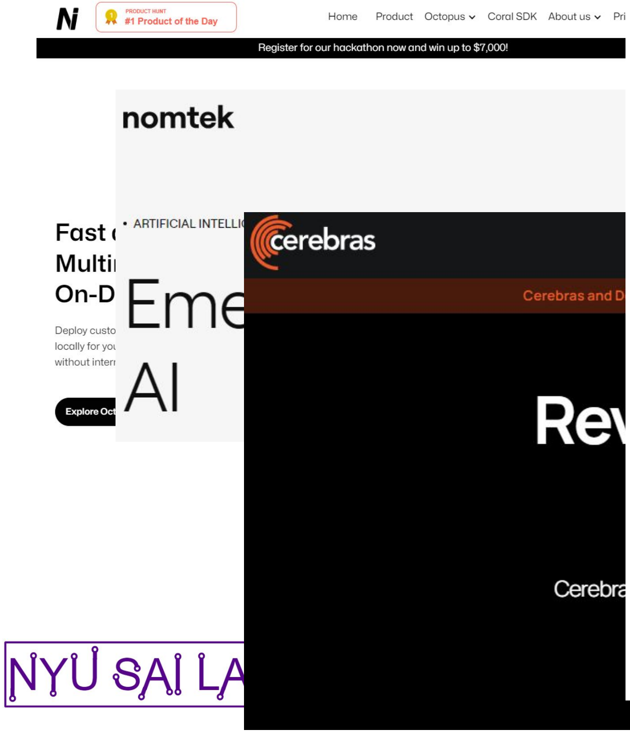
gree.

alent practical experience.

kaging).

Jc

AI Tech Startups/Unicorns



PRODUCT HUNT
#1 Product of the Day

Home Product Octopus Coral SDK About us Pri

Register for our hackathon now and win up to $7,000!

Fast (
Multi
On-D

Deploy custo
locally for you
without inter

Explore Oct

nomtek

• ARTIFICIAL INTELLI

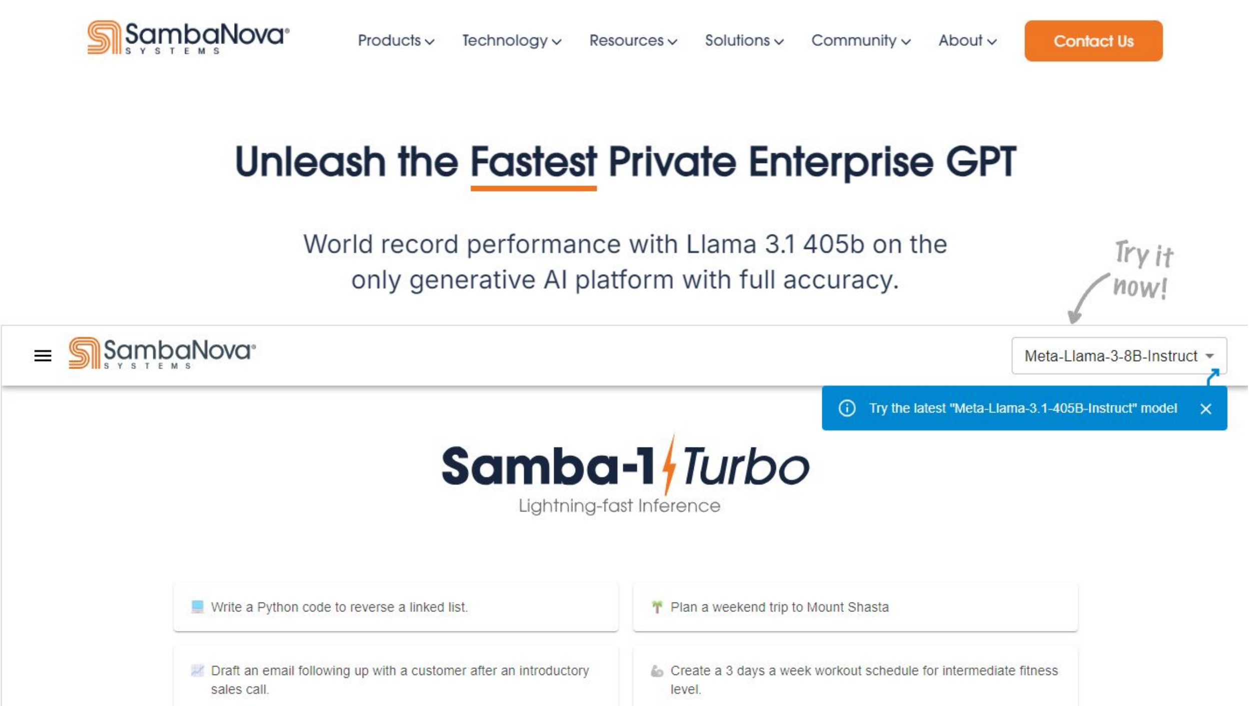
Eme
AI

cerebras

Cerebras and D

Rev

Cerebra



SambaNova SYSTEMS

Products Technology Resources Solutions Community About Contact Us

Unleash the Fastest Private Enterprise GPT

World record performance with Llama 3.1 405b on the only generative AI platform with full accuracy.

Try it now!

SambaNova SYSTEMS

Meta-Llama-3-8B-Instruct

Try the latest "Meta-Llama-3.1-405B-Instruct" model

Samba-1 Turbo
Lightning-fast Inference

Write a Python code to reverse a linked list.

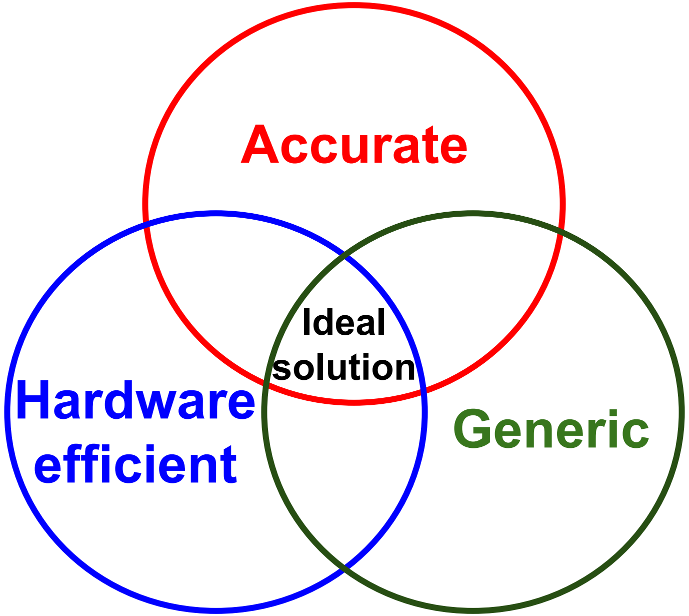
Plan a weekend trip to Mount Shasta

Draft an email following up with a customer after an introductory sales call.

Create a 3 days a week workout schedule for intermediate fitness level.

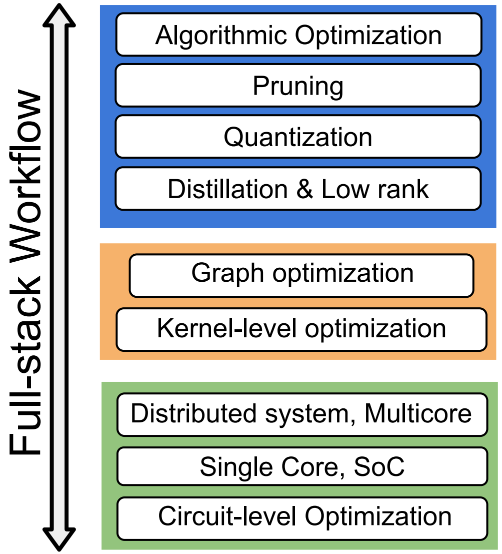
Efficient AI: An Emerging Area

Challenges



NYU SAI LAB

Efficient AI: Full-stack Workflow



Full-stack Workflow

Algorithmic Optimization

Pruning

Quantization

Distillation & Low rank

Graph optimization

Kernel-level optimization

Distributed system, Multicore

Single Core, SoC

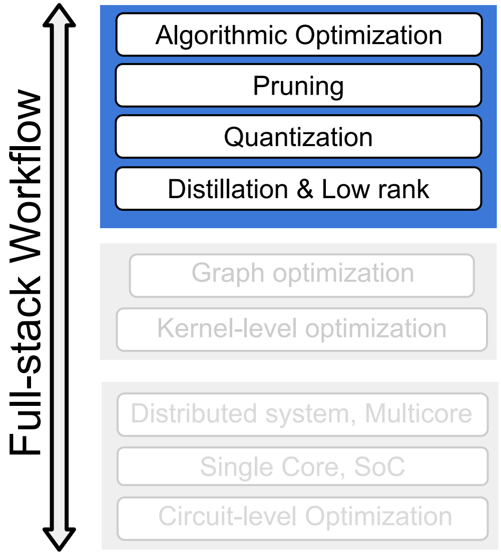
Circuit-level Optimization

Efficient Algorithm

DNN Compiler

DNN Hardware Accelerator

Efficient AI: Full-stack Workflow



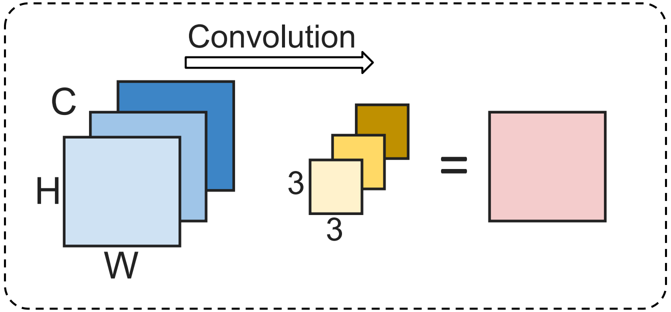
Efficient Algorithm

DNN Compiler

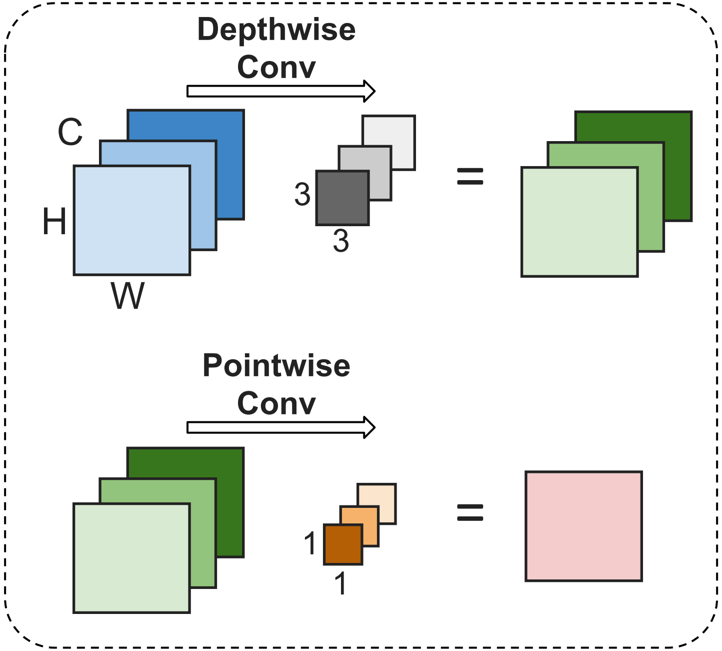
DNN Hardware Accelerator

Algorithmic Optimization

Standard Convolution

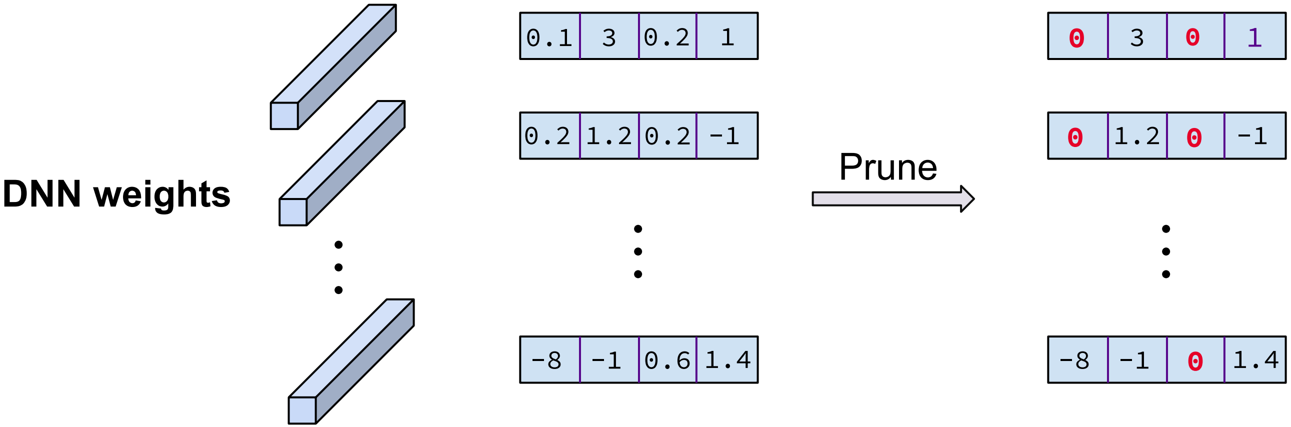


Depthwise Separable Convolution



NYU SAI LAB

Efficient DNN Algorithm: Pruning



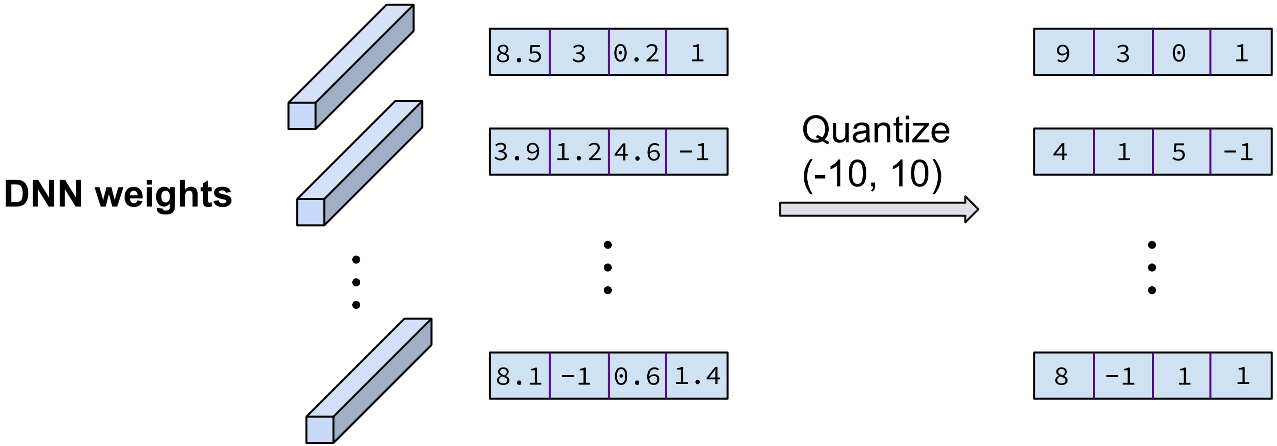
DNN weights

0.1	3	0.2	1
0.2	1.2	0.2	-1
⋮			
-8	-1	0.6	1.4

Prune

0	3	0	1
0	1.2	0	-1
⋮			
-8	-1	0	1.4

Efficient DNN Algorithm: Quantization



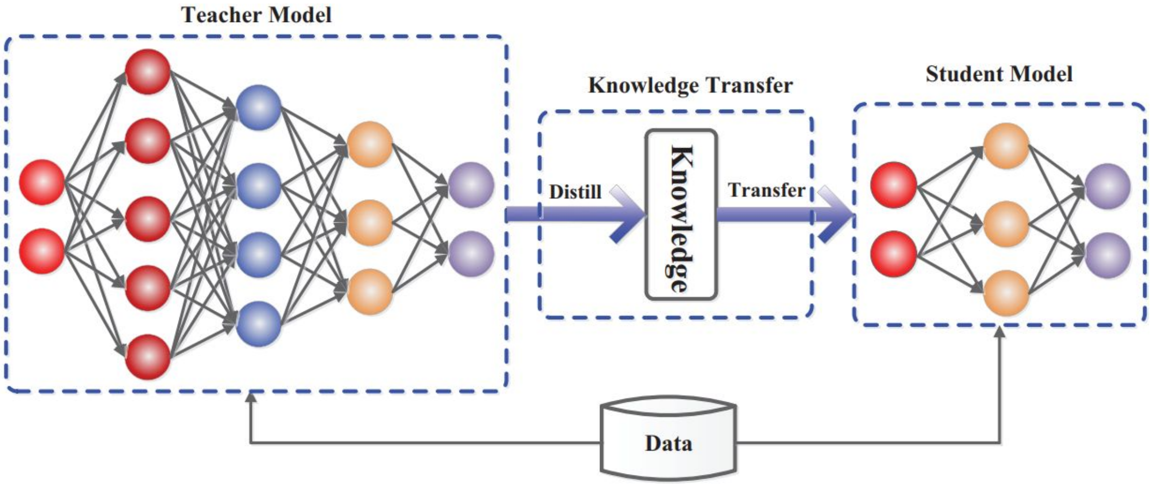
DNN weights

8.5	3	0.2	1
3.9	1.2	4.6	-1
⋮			
8.1	-1	0.6	1.4

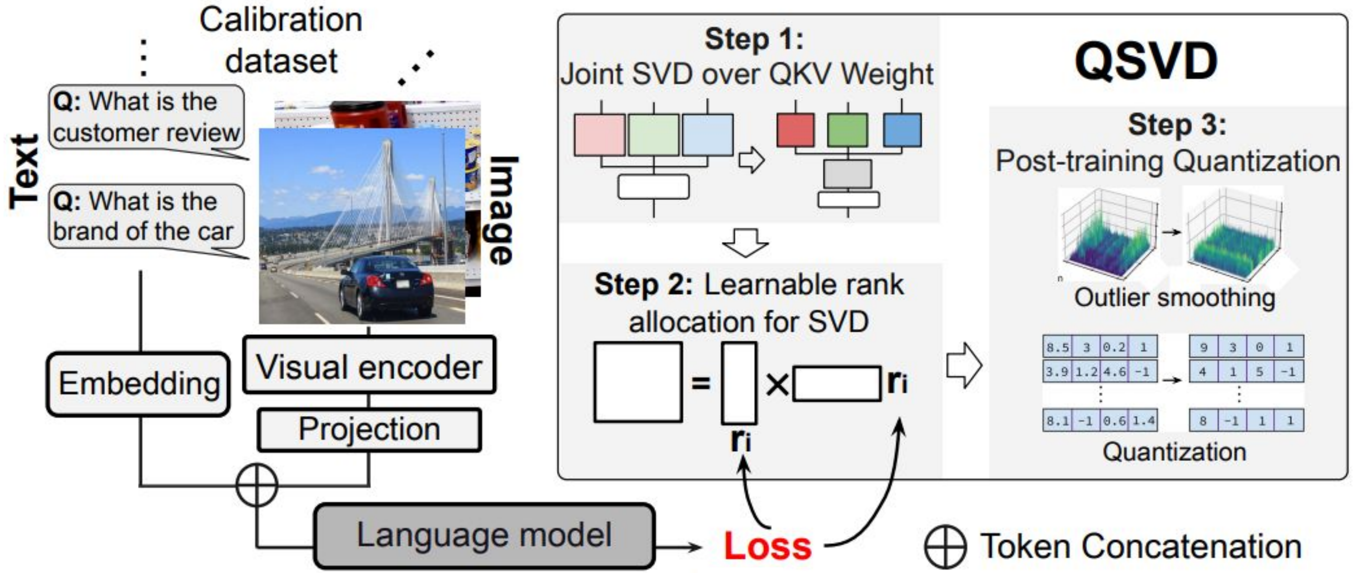
Quantize (-10, 10)

9	3	0	1
4	1	5	-1
⋮			
8	-1	1	1

Knowledge Distillation



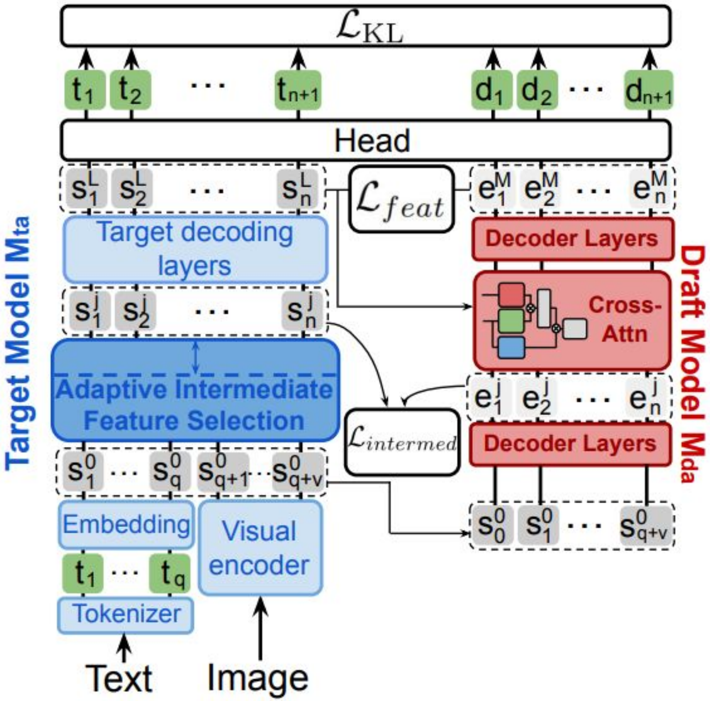
QSVD

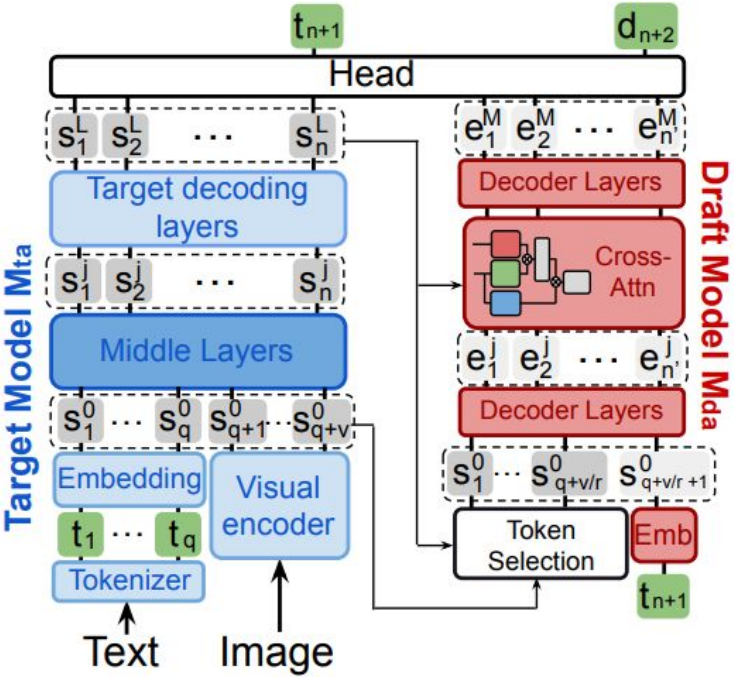


- We propose leveraging Singular-Value Decomposition over the joint query (Q), key (K), and value (V) weight matrices to reduce KV cache size and computational overhead.

NYU SAI LAB

Speculative Decoding with DREAM

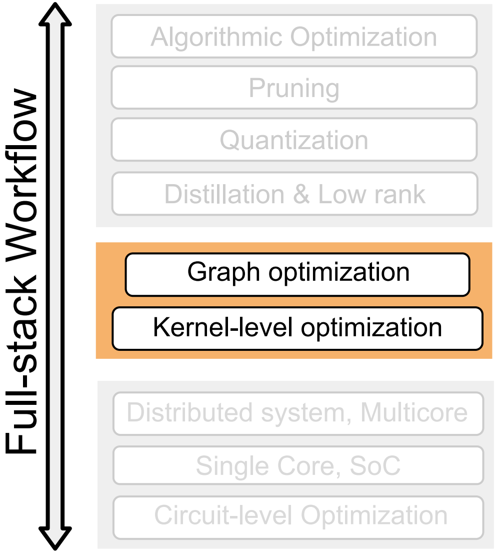

$\mathcal{L}_{\mathrm{KL}}$
t_1 t_2 ··· t_{n+1}
d_1 d_2 ··· d_{n+1}
Head
s_1^L s_2^L ··· s_n^L
$\mathcal{L}_{feat}$
e_1^M e_2^M ··· e_n^M
Target decoding layers
Decoder Layers
Cross-Attn
s_1^j s_2^j ··· s_n^j
Adaptive Intermediate Feature Selection
e_1^j e_2^j ··· e_n^j
$\mathcal{L}_{intermed}$
Decoder Layers
s_1^0 ··· s_q^0 s_{q+1}^0 ··· s_{q+v}^0
s_0^0 s_1^0 ··· s_{q+v}^0
Embedding
Visual encoder
t_1 ··· t_q
Tokenizer
Text
Image
Target Model Mta
Draft Model Mda



NYU SAI LAB

- We introduce DREAM, a novel speculative decoding framework tailored for VLMs.

Efficient AI: Full-stack Workflow



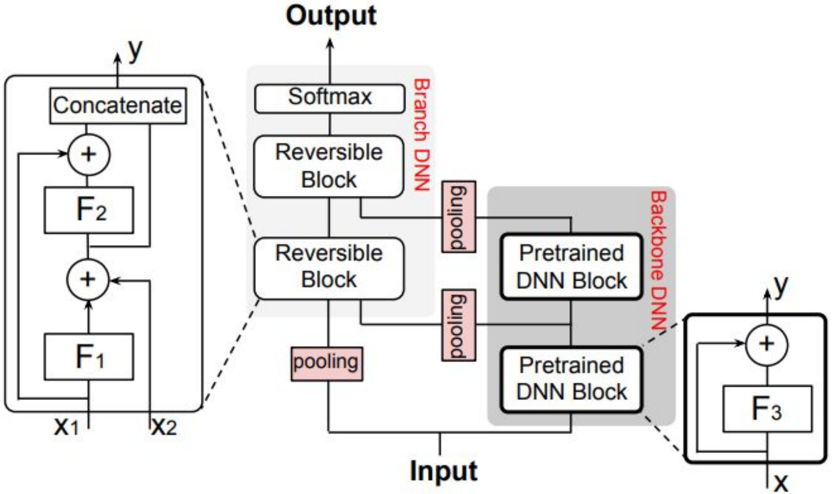
Efficient Algorithm

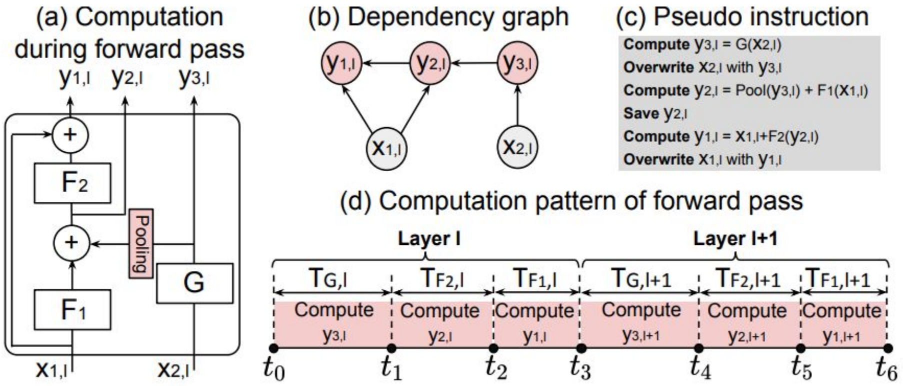
DNN Compiler

DNN Hardware Accelerator

Graph Level Optimization

CAMEL Training

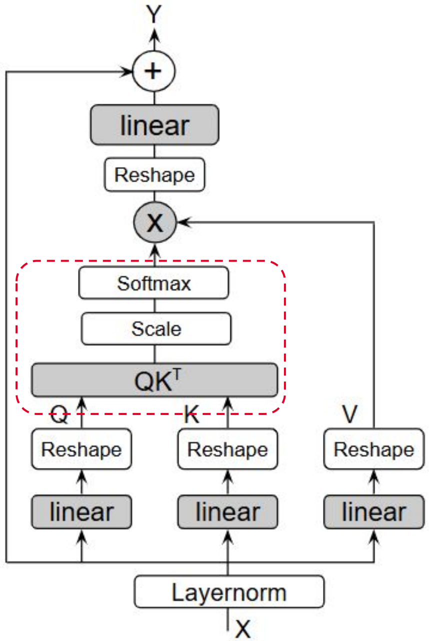


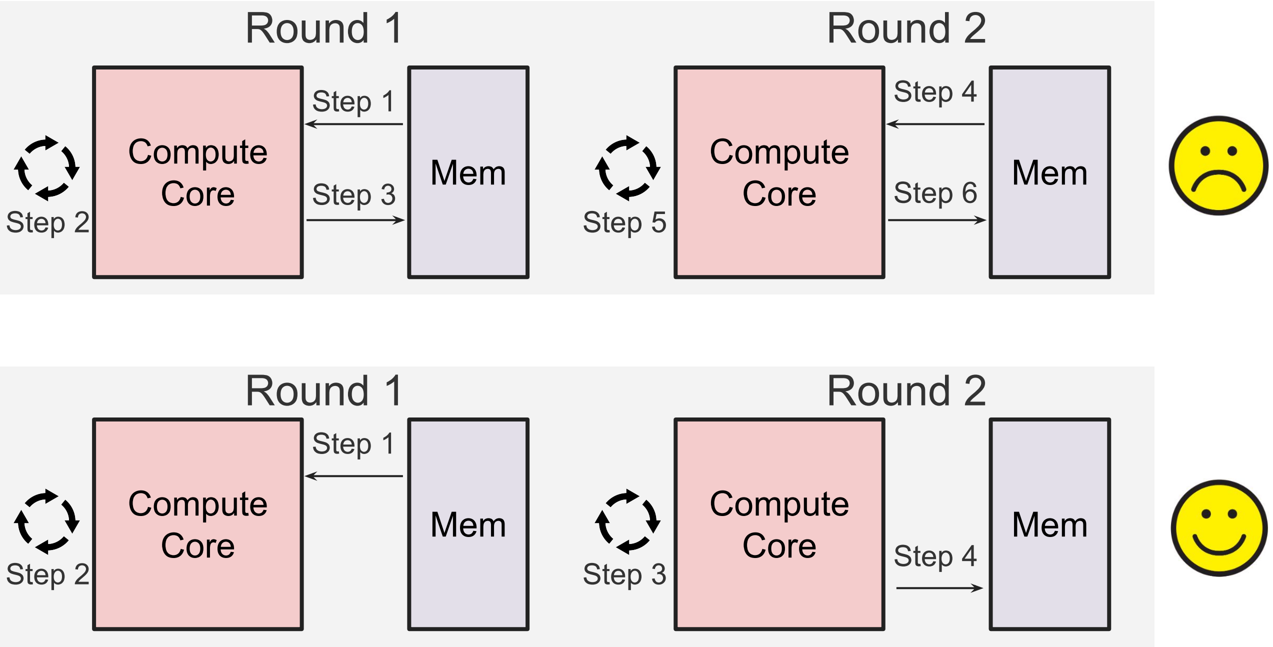


```
Compute y3,l = G(x2,l)
Overwrite x2,l with y3,l
Compute y2,l = Pool(y3,l) + F1(x1,l)
Save y2,l
Compute y1,l = x1,l+F2(y2,l)
Overwrite x1,l with y1,l
```

Zhang, Sai Qian, et al. "Camel: Co-designing ai models and edrams for efficient on-device learning." *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024.

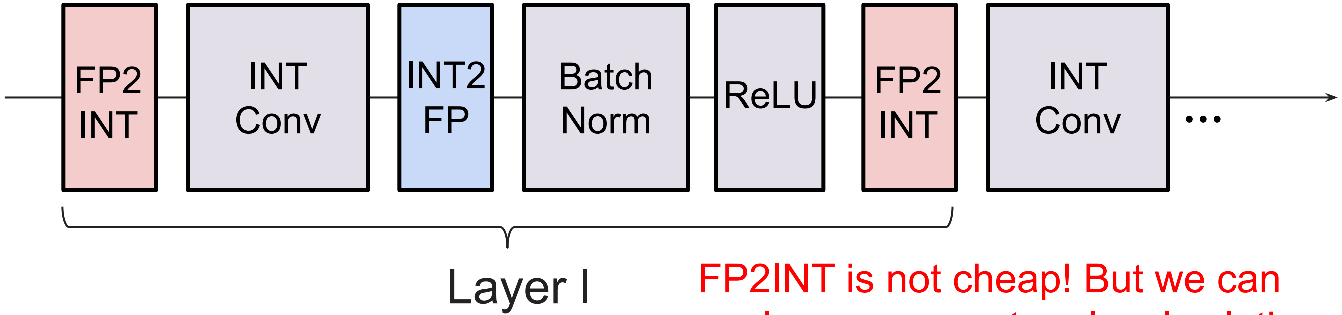
System Level Optimization





System Level Optimization

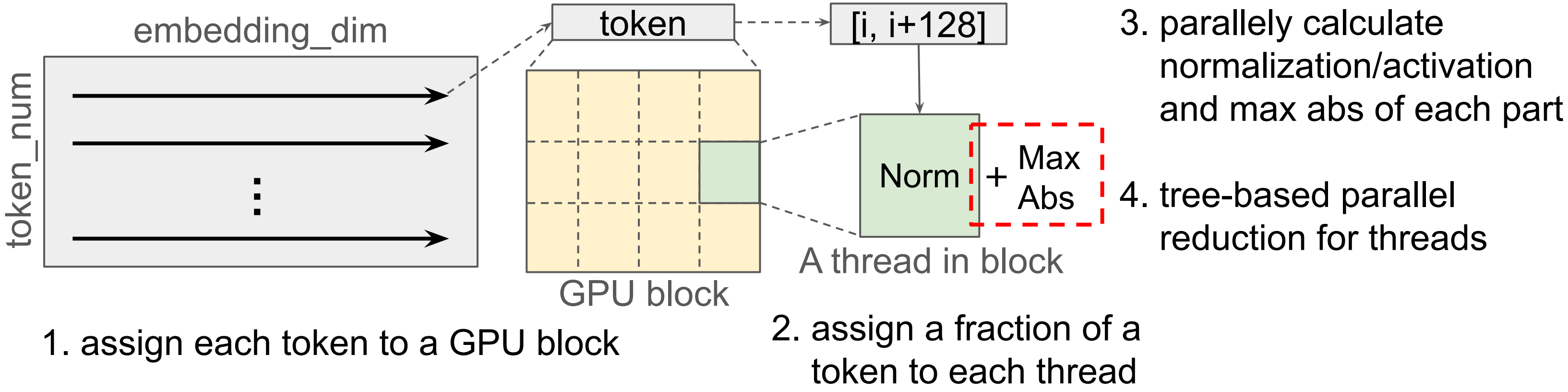
- How to convert a number x to INT representation?
  - Set the clipping range: (-L, L), bitwidth: b
  - Compute the scale: $s = 2L/(2^b - 2)$
  - Clip the input x: $x_c = Clip(x, L, -L)$
  - Calculate the INT representation: $x_{int} = round(x_c/s)$
  - Rescale: $x_q = sx_{int}$



FP2INT is not cheap! But we can explore some system-level solution

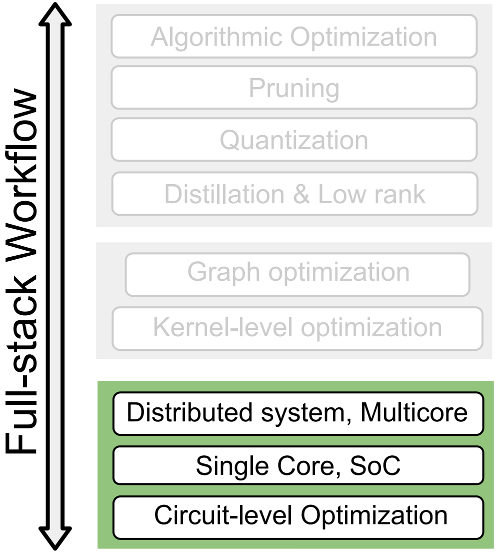
NYU SAI LAB

Kernel Fusion


embedding_dim
token_num
token
[i, i+128]
Norm
+
Max
Abs
GPU block
A thread in block
1. assign each token to a GPU block
2. assign a fraction of a token to each thread
3. parallely calculate normalization/activation and max abs of each part
4. tree-based parallel reduction for threads

- For example, we can fuse the max searching operation to the batch normalization operation within LLM.

Efficient AI: Full-stack Workflow



Efficient Algorithm

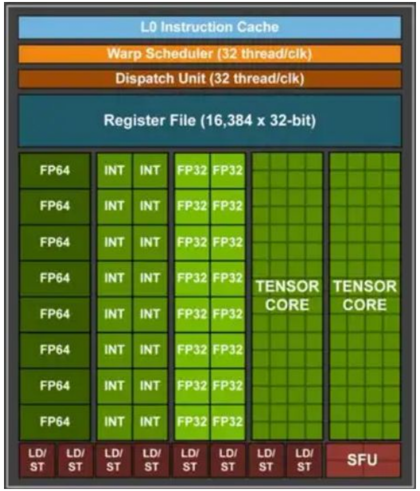
DNN Compiler

DNN Hardware Accelerator

NYU SAI LAB

Hardware Support for DNN

- GPU is better than CPU in terms of throughput for both Neural Network training and inference.
  - GPU leverages the highly parallelized architecture of its computing units to handle computational intensive operations.
- However, GPU:
  - General purpose, although much more specific than CPU.
  - Still not fast and power-efficient enough.
  - Does not support advanced efficient DNN algorithm.



L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
FP64
INT INT
FP32 FP32
TENSOR CORE
TENSOR CORE
LD/ST
SFU

NVIDIA

NVIDIA H100

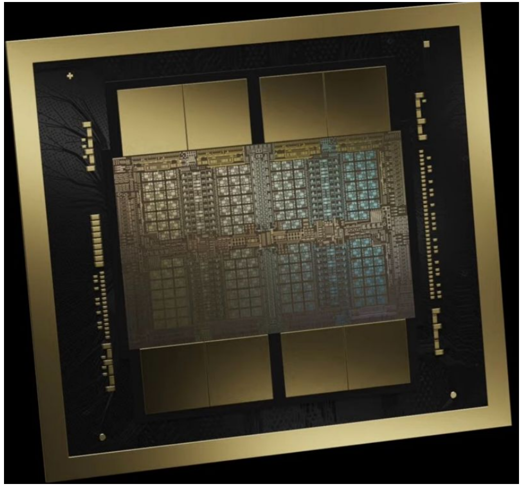
Chip size	814 mm^2
On-chip memory	~50MB
Total memory	~96GB HBM
Cores	16,896 FP32 + 528 Tensor
Precision	FP16/FP8/INT8
Memory bandwidth	0.003 Petabytes/sec

NYU SAI LAB

https://www.techpowerup.com/gpu-specs/h100-sxm5-96-gb.c3974

NVIDIA

Chip size	-
On-chip memory	-
Total memory	192GB HBM
Cores	-
Precision	FP16/FP8/FP4/INT8
Memory bandwidth	8 Terabytes/sec



NVIDIA Blackwell

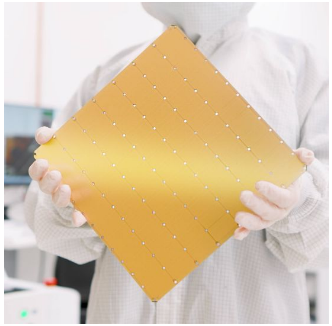
https://wccftech.com/nvidia-blackwell-gpu-architecture-official-208-billion-transistors-5x-ai-performance-192-gb-hbm3e-memory/

Hardware Support for DNN

- ASIC-based implementations have been recently explored to accelerate the DNN inference.
  - Google’s TPU, Apple’s Neural Engine, Cerebras AI chip, …
- FPGA-based accelerators for DNN inference have been recently developed.
  - Has good programmability and flexibility
  - Short development cycles
  - Can be used as a benchmark before implementing on ASIC

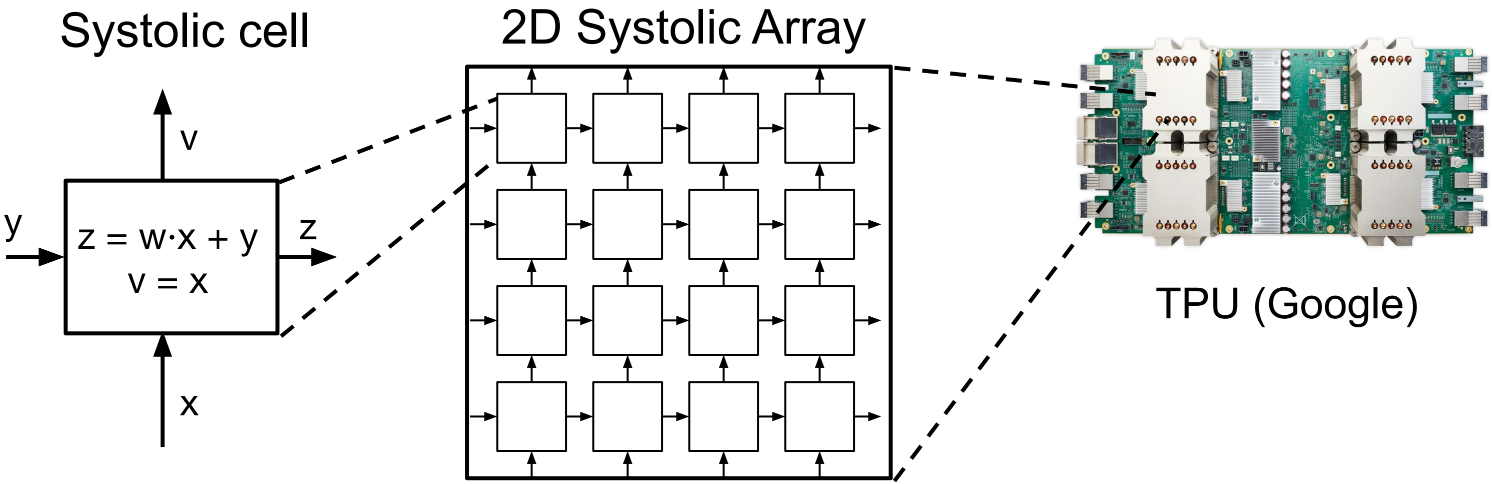
Tensor Processing Unit (Google)

Alveo Accelerator Card (Xilinx)


Cerebras CS-3

Systolic Array

- Kung and Leiserson, "Systolic Arrays for VLSI," 1978 and Kung, "Why systolic architectures?' 1982
- 2D grid of multiplier-accumulators (MACs) for matrix multiplication
- Used by Google TPU for deep learning (2017), etc



Systolic cell

2D Systolic Array

TPU (Google)

Bit-serial Low-precision Multiplier

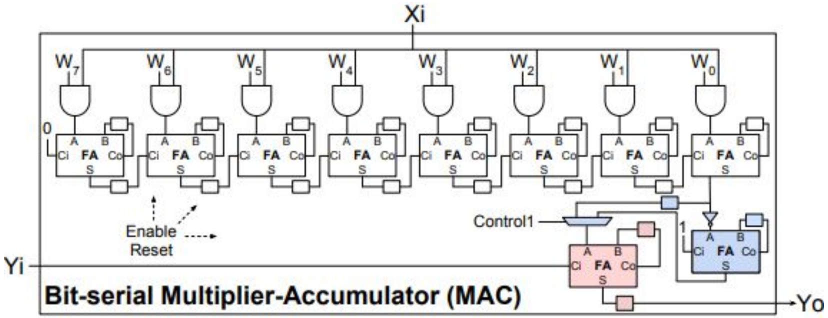
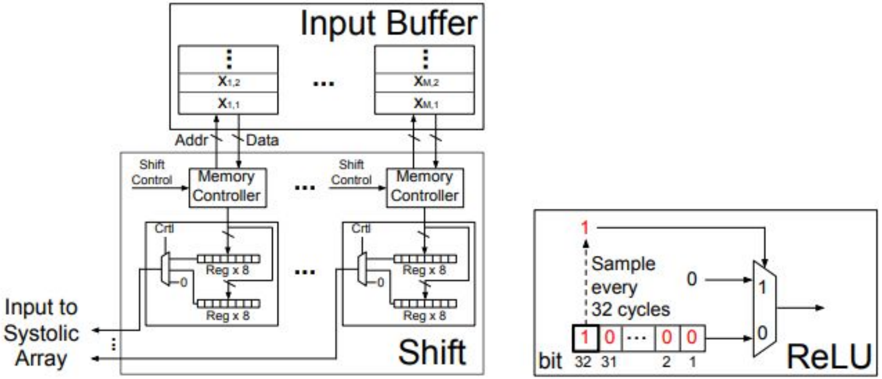
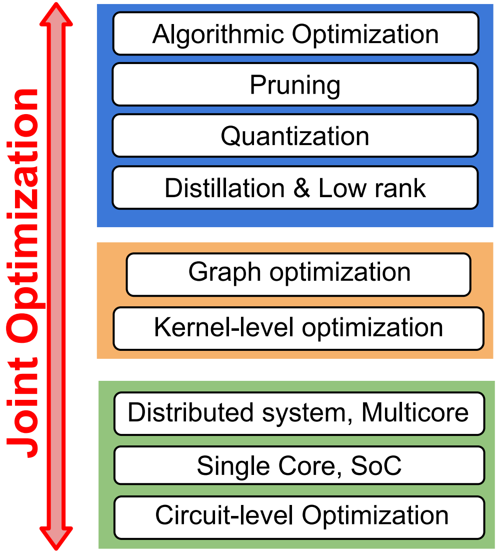


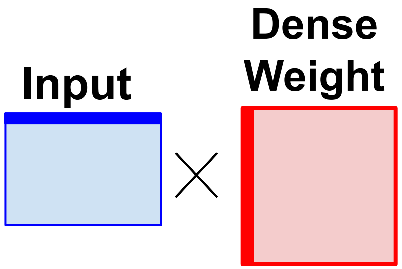
Figure 7: Bit-serial multiplier-accumulator (MAC).



Why We Need Codesign?

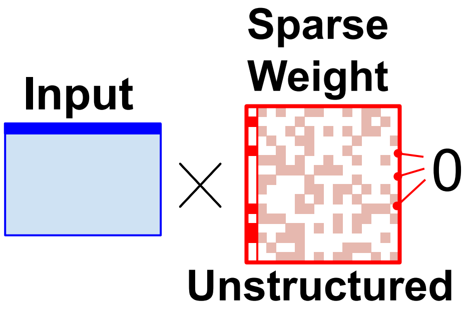


Why We Need Codesign?



Input

Dense Weight



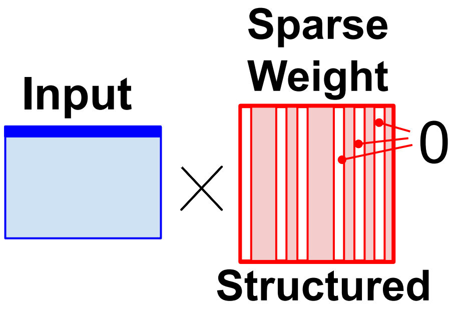
Input

Sparse Weight

0

Unstructured

High accuracy
low hardware efficiency



Input

Sparse Weight

0

Structured

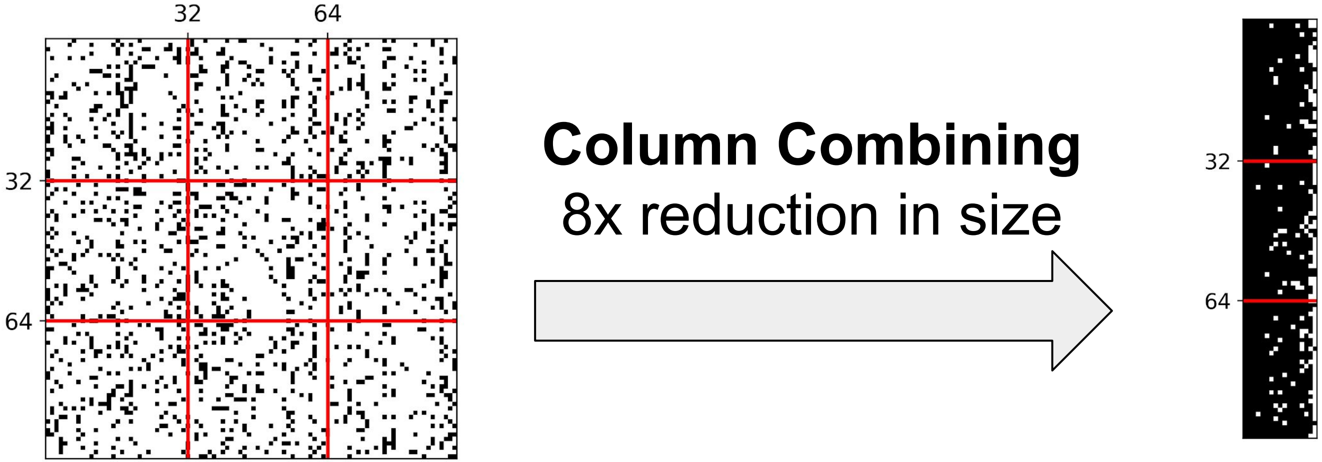
Low accuracy
high hardware efficiency

Hardware architecture needs to be considered when designing efficient DNN.

Column Combining

Sparse
Weight Matrix

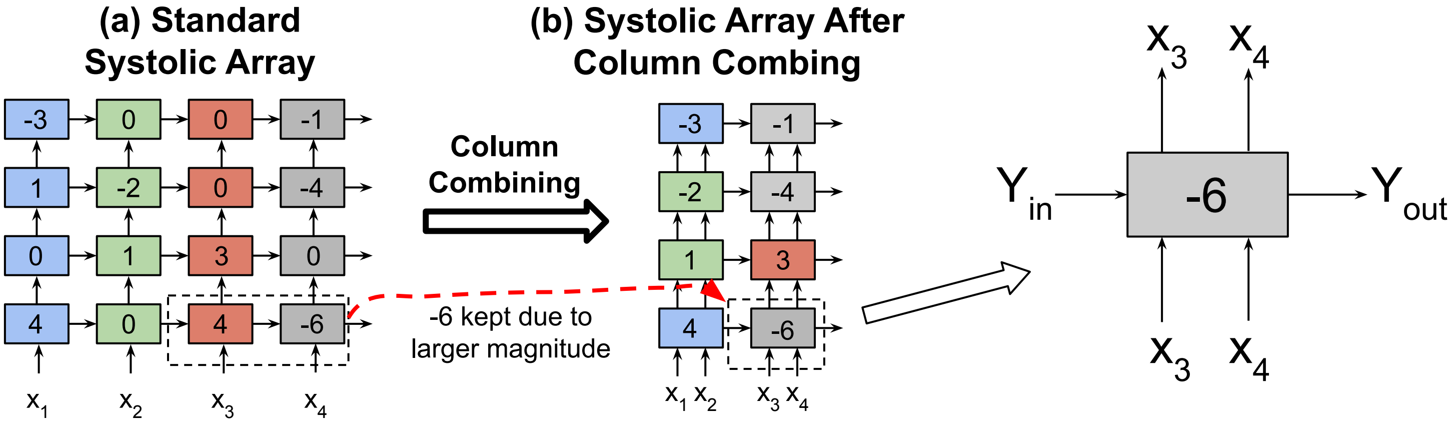
Packed Format in
Systolic Array



Column Combining
8x reduction in size

Kung, H. T., Bradley McDanel, and Sai Qian Zhang. "Packing sparse convolutional neural networks for efficient systolic array implementations: Column combining under joint optimization." *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 2019.

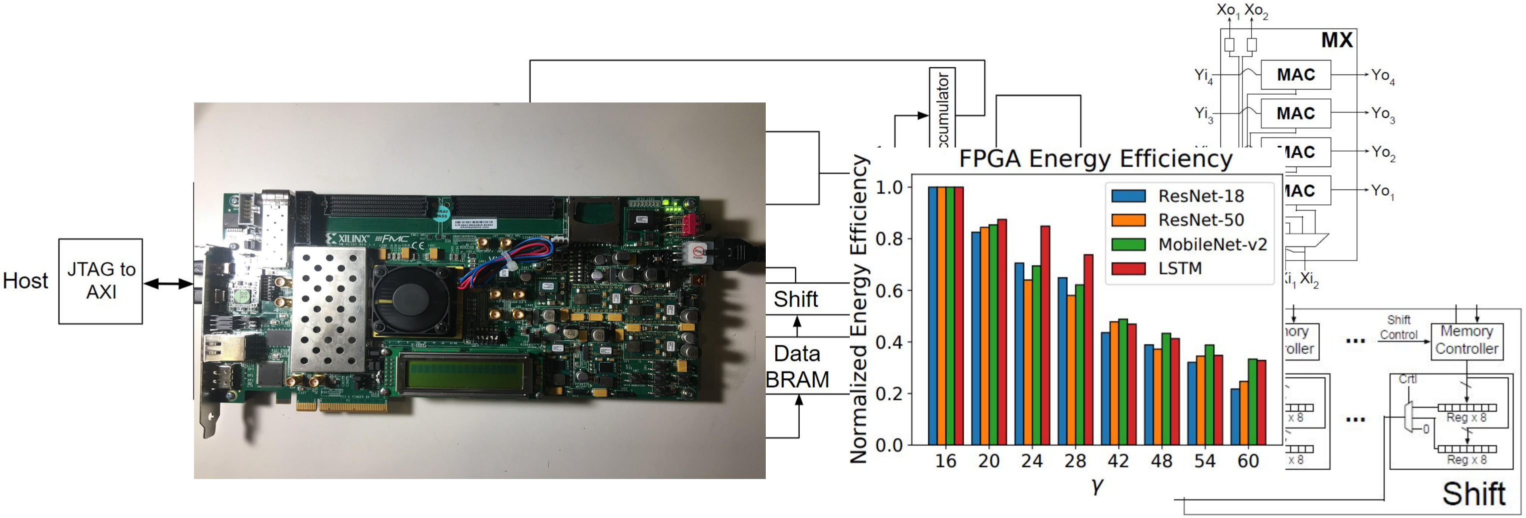
Column Combining



- Column combining can greatly increase the utilization efficiency of the systolic array
- Recently, Nvidia A100 GPU adopts a similar idea to support the balanced structured sparsity on their GPU

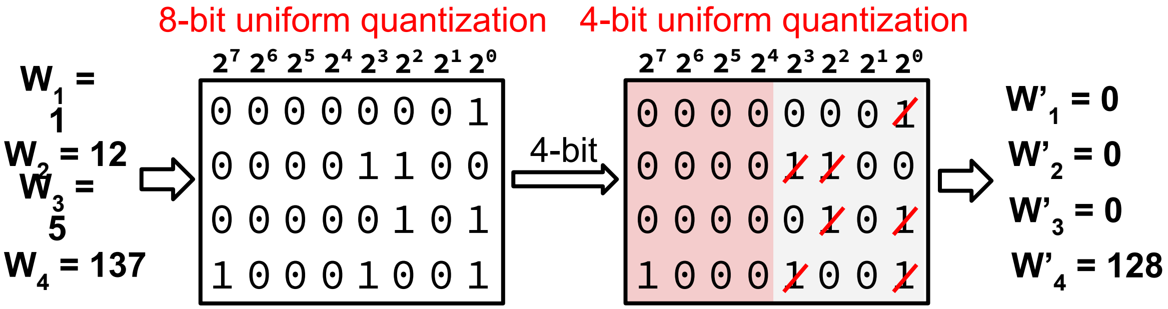
Kung, H. T., Bradley McDanel, and Sai Qian Zhang. "Packing sparse convolutional neural networks for efficient systolic array implementations: Column combining under joint optimization." *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 2019.

FPGA Accelerator



Kung, H. T., Bradley McDanel, and Sai Qian Zhang. "Packing sparse convolutional neural networks for efficient systolic array implementations: Column combining under joint optimization." *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 2019.

Term Quantization



8-bit uniform quantization

$W_1 = 1$

$W_2 = 12$

$W_3 = 5$

$W_4 = 137$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	1
0	0	0	0	1	1	0	0
0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	1

4-bit

4-bit uniform quantization

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	~~1~~
0	0	0	0	~~1~~	~~1~~	0	0
0	0	0	0	0	~~1~~	0	~~1~~
1	0	0	0	~~1~~	0	0	~~1~~

$W'_1 = 0$

$W'_2 = 0$

$W'_3 = 0$

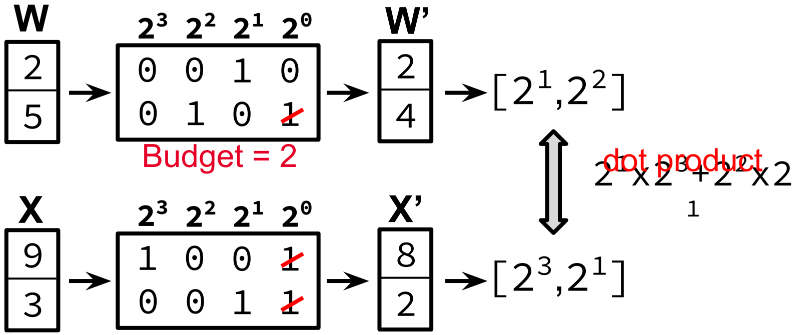
$W'_4 = 128$

- Low-precision quantization leads to significant quantization error.
- Both weights and input activation are highly biased in values.

NYU SAI LAB

Kung, H. T., Bradley McDanel, and Sai Qian Zhang. "Term revealing: Furthering quantization at run time on quantized dnns." *arXiv preprint arXiv:2007.06389* (2020).

Term Quantization



4-bit uniform quantization

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	1
0	0	0	0	1	1	0	0
0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	1

$W'_1 = 0$

$W'_2 = 0$

$W'_3 = 0$

$W'_4 = 128$

TQ with a budget = 4

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0	0	1
0	0	0	0	1	1	0	0
0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	1

$W'_1 = 0$

$W'_2 = 12$

$W'_3 = 0$

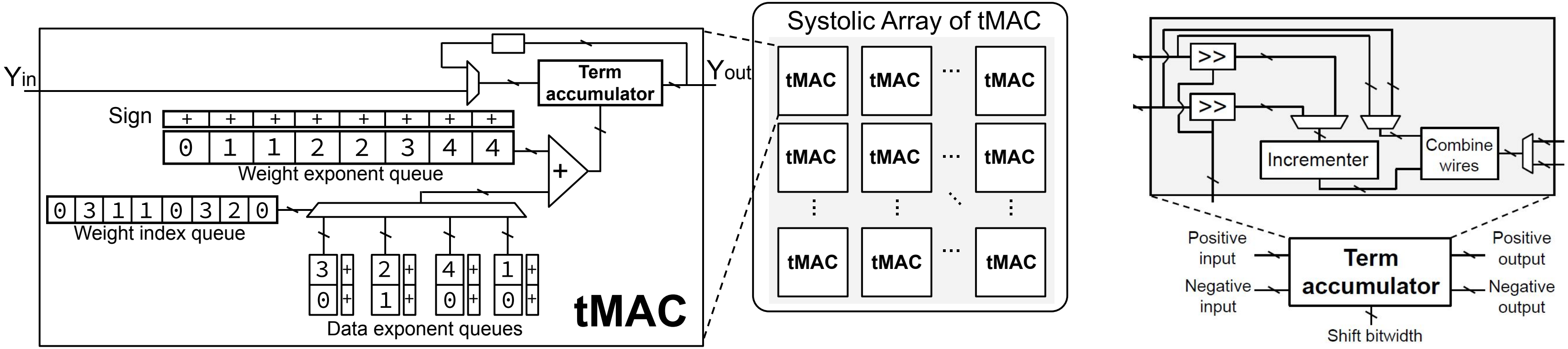
$W'_4 = 136$

- We can control the term-level computations by setting a **group term budget.**
- For a group of values, we rank and remove the small terms based on this budget.

NYU SAI LAB

Kung, H. T., Bradley McDanel, and Sai Qian Zhang. "Term revealing: Furthering quantization at run time on quantized dnns." *arXiv preprint arXiv:2007.06389* (2020).

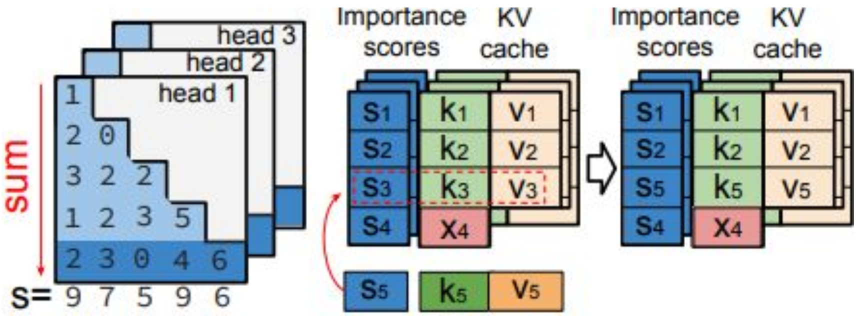
Term Quantization: Accelerator Design

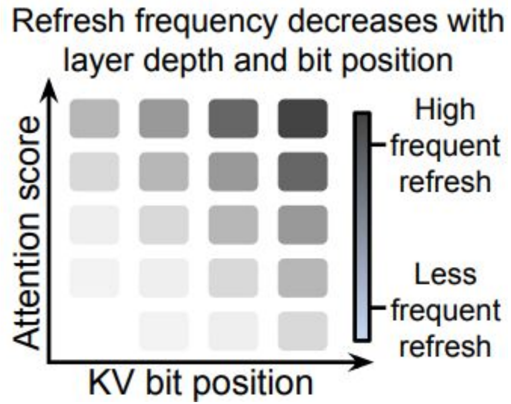


- We propose the term MAC (tMAC) for the efficient implementation of TQ.
- A tMAC processes all term-pair multiplications across a group of weight and data values.
- Each term is represented by their corresponding exponent (2-3 bits).
- The term accumulation can be implemented using half adders.

Kung, H. T., Bradley McDanel, and Sai Qian Zhang. "Term revealing: Furthering quantization at run time on quantized dnns." *arXiv preprint arXiv:2007.06389* (2020).

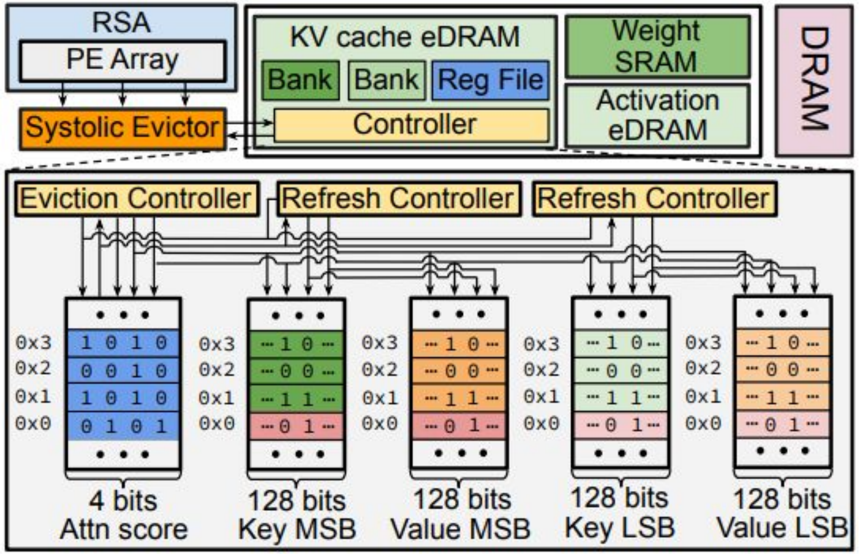
Kelle: Co-design KV Caching and eDRAM for Efficient LLM Serving in Edge Computing





- We propose using embedded DRAM (eDRAM) as the primary storage for LLM serving in edge device, which offers higher storage density compared to SRAM.
- To reduce eDRAM costs and improve overall system performance, we propose Kelle, a software-hardware co-design solution optimized for deploying LLMs on eDRAMbased edge systems.

Kelle: Co-design KV Caching and eDRAM for Efficient LLM Serving in Edge Computing



- Combined with our fine-grained memory eviction, recomputation, and refresh control algorithms, the Kelle accelerator delivers a 3.9× speedup and 4.5× energy savings compared to existing baseline solutions.

NYU SAI LAB

Lecture Plan (Tentative)

Chapter 1: Basics and Efficient DNN Architectures

- Lecture 1: Review the basics of DNN
- Lecture 2: CNNs, RNNs and Variants
- Lecture 3: Transformer and its Application in AIGC

Lecture Plan (Tentative)

Chapter 2: Efficient DNN Algorithms

- Lecture 4: DNN Pruning
- Lecture 5: DNN Quantization
- Lecture 6: Distillation, Low rank Decomposition and NAS
- Lecture 7: Algorithm for Large Model Efficiency
- Lecture 8: Efficient DNN Training, Distributed Training, Federated Learning

Lecture Plan (Tentative)

Chapter 3: System and Hardware Design for AI

- Lecture 9: Distributed Machine Learning System for Training and Inference
- Lecture 10: Machine Learning System for Large Model
- Lecture 11: AI Accelerator Introduction and CNN Accelerators
- Lecture 12: Transformer & LLM Accelerators
- Lecture 13: The Future of Efficient AI
    - Guest Lecture: Vithursan Thangarasa (Cerebras)