Embedding Operations in Deep Learning Recommendation Models

Jongsoo Park, Research Scientist, Facebook AI System Co-design
w/ Jianyu Huang, Andrew Tulloch, Xing Liu, Jie (Amy) Yang, Mustafa Ozdal, Dheevatsa Mudigere, and other contributors

Mar 2021

Outline

- Deep Learning Recommendation Models (DLRM)
- Deep Dive in DLRM Embedding Operations

Outline

- **Deep Learning Recommendation Models (DLRM)**
- Deep Dive in DLRM Embedding Operations

DLRM is a part of MLPerf

Neural Machine Translations

Accessibility

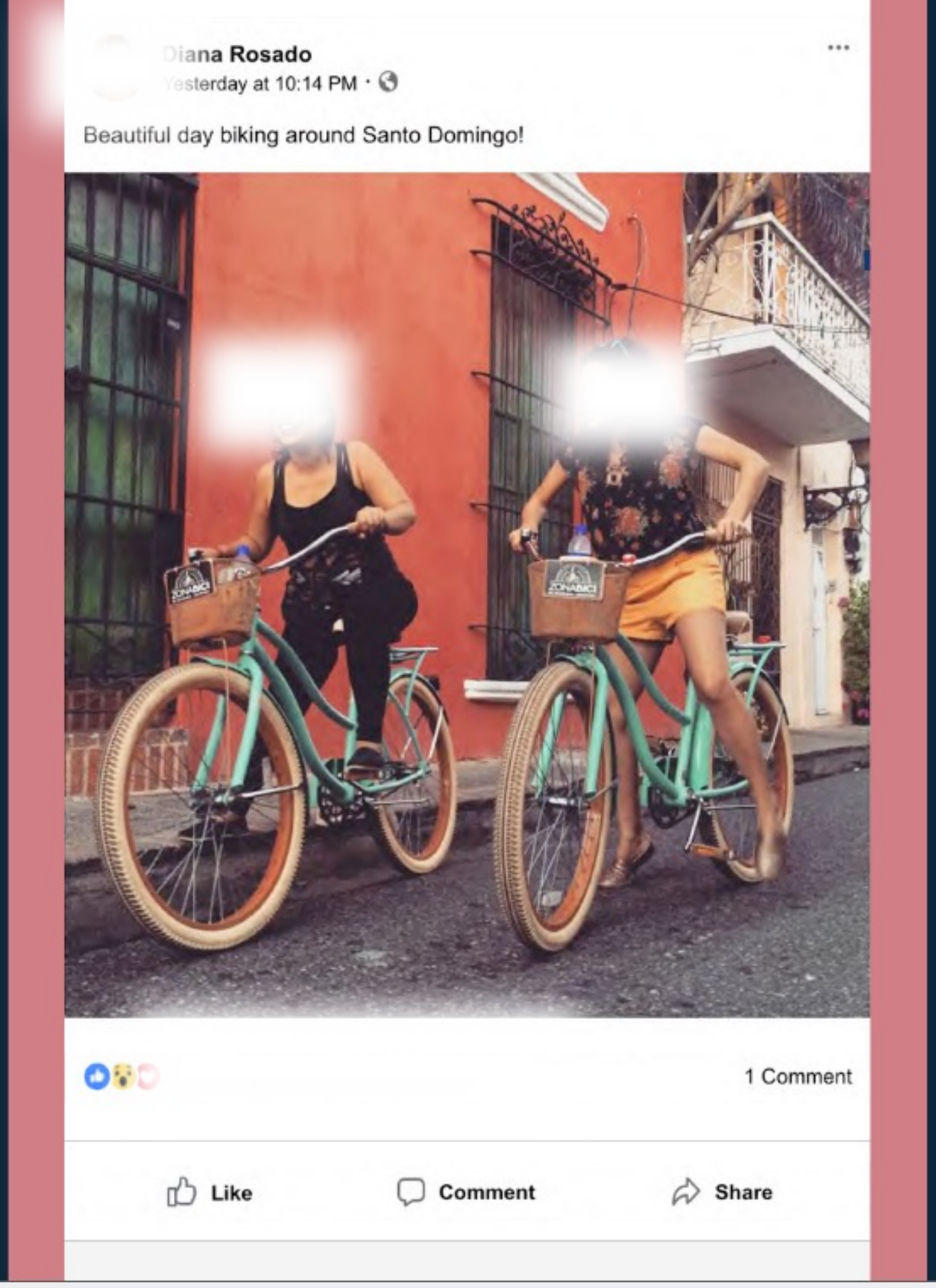


IMAGE CONTAINS

Diana & Hanna on

Riding Bikes in

Biking Near a Red Building

AI @ Facebook

- Srinivas Narayanan, Going Beyond Fully Supervised

AI growth and scale @ Facebook

ML data growth

Usage in 2018: **30%**

Usage today: **50%**

Growth in one year: **3X**

1-year training growth

Ranking engineers: **2X**

Workflows trained: **3X**

Compute consumed: **3X**

Inference scale per day

# of predictions: **400T**

# of translations: **6.5B**

Fake accounts removed: **99%**

Naumov and Mudiger, Recommendation Systems using DLRM, KDD PyTorch workshop, 2020

What are the workloads?

- **Ranking and recommendation**

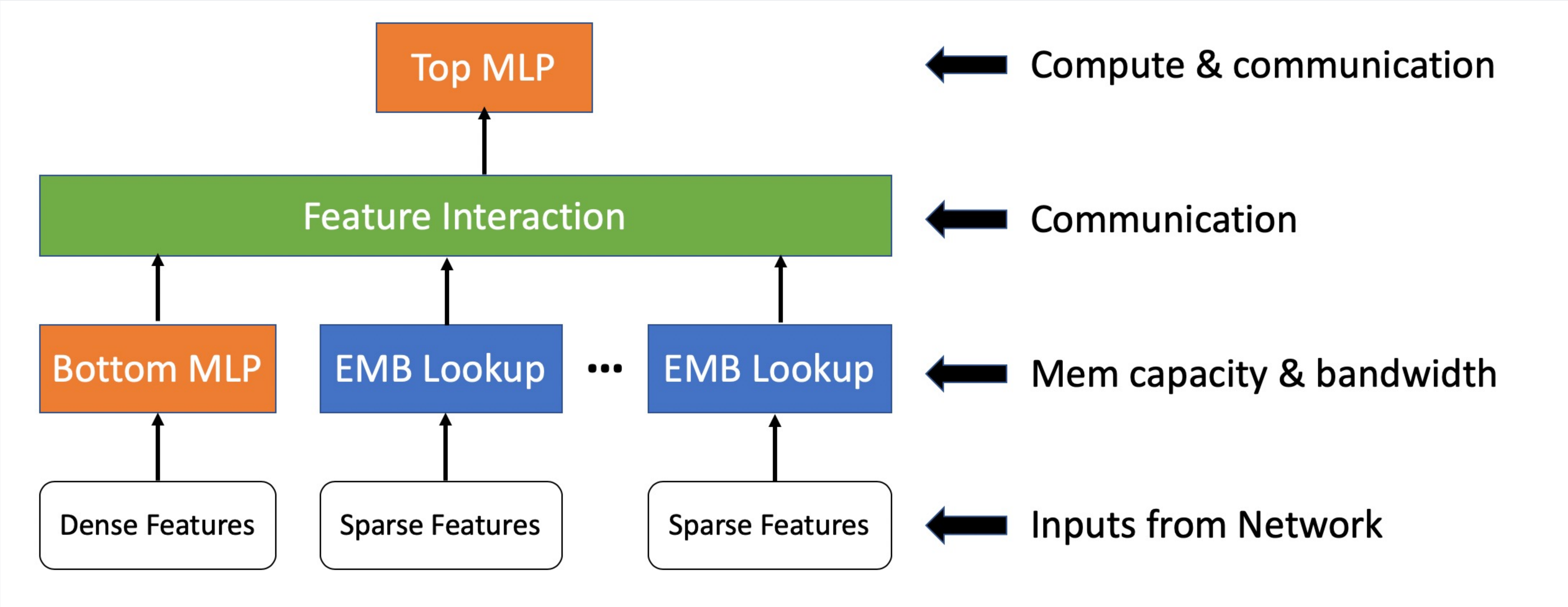
- Computer vision

  Image classification, object detection, and video understanding

- Language

  Translation, speech recognition, content understanding

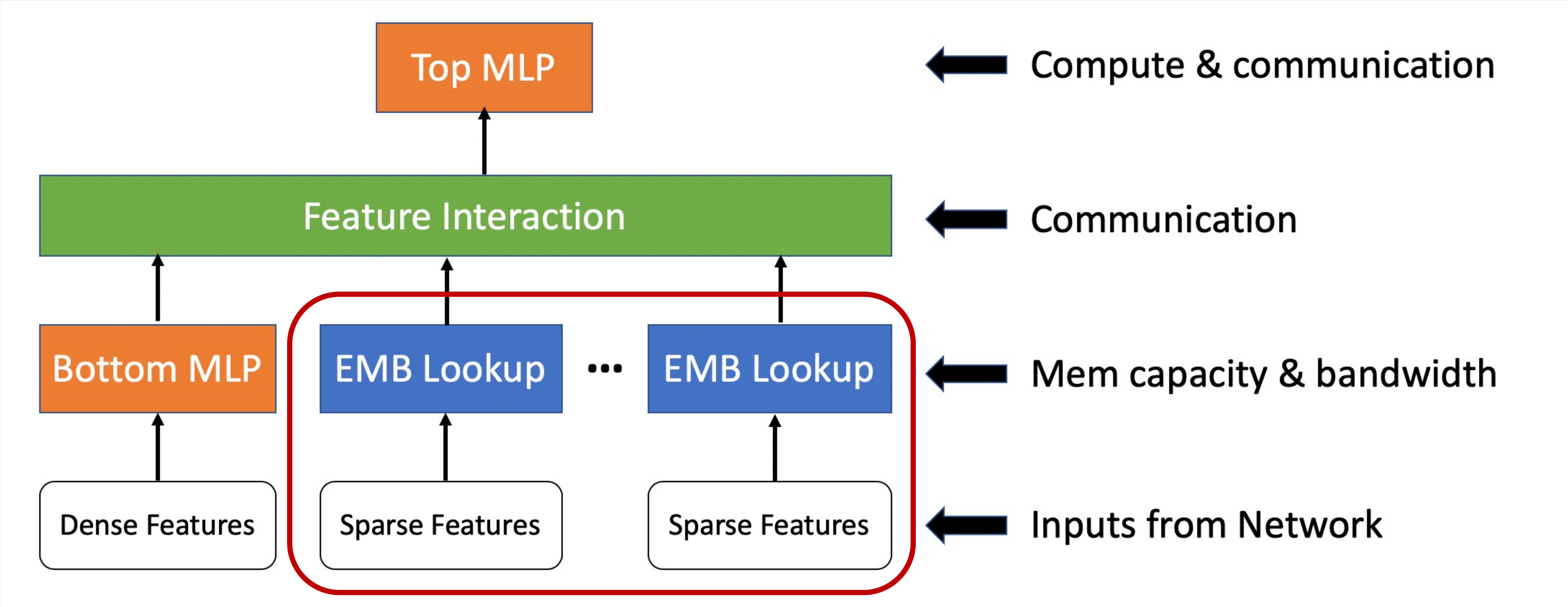
Deep Learning Recommendation Model (DLRM)



Deep Learning Recommendation Model for Personalization and Recommendation Systems, Naumov et al., 2019

https://github.com/facebookresearch/dlrm

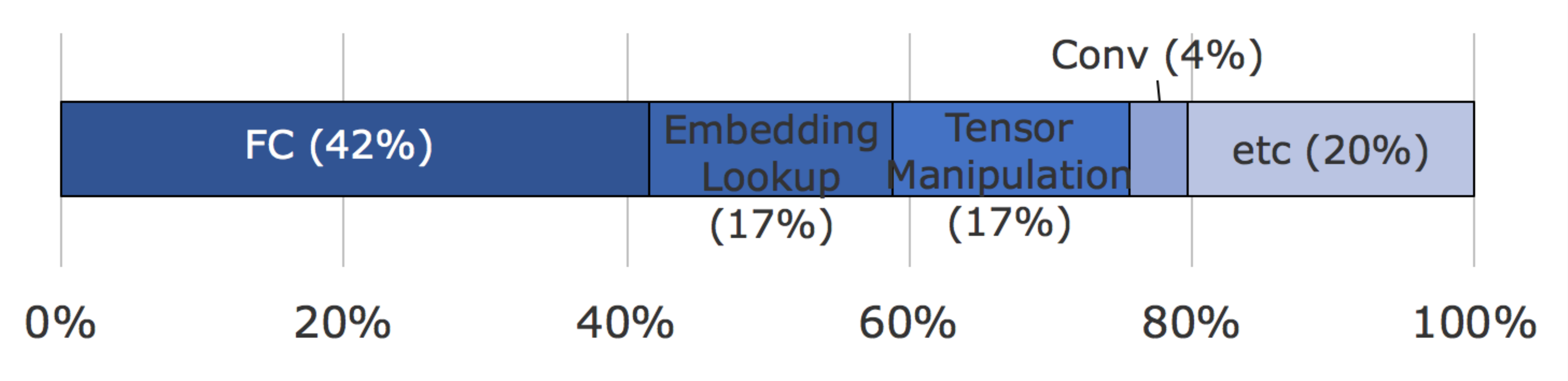
Deep Learning Recommendation Model (DLRM)



Deep Learning Recommendation Model for Personalization and Recommendation Systems, Naumov et al., 2019

https://github.com/facebookresearch/dlrm

Fleet-wide DL inference execution time breakdown



- FC is the most time consuming followed by **embedding** (from recommendation models)

Deep Learning Inference in Facebook Data Centers: Characterization, Performance Optimizations and Hardware Implications, Park et al., 2018

Outline

- Deep Learning Recommendation Models (DLRM)
- **DLRM Embedding Operations**

Don’t worry about details! We open sourced actual implementation at https://github.com/pytorch/FBGEMM

Types of embedding operations

- Forward: EmbeddingBag in PyTorch, SparseLengthsSum in Caffe2
- EmbeddingBag bwd fused with sparse optimizers (AdaGrad, SGD, Adam, LAMB, …)

Embedding Bag = Sparse x Dense Matmul

Embedding Table

FBGEMM → 1 1 1 **X**

Efficient	X_{11}
Low-Precision	X_{23}
Inference	X_{99}

FBGEMM's Embedding = $X_{11} + X_{23} + X_{99}$

Optimization goal priorities

Accuracy

>> scalability and memory size

> single device speed

Challenge 1. Memory capacity and BW demand

- ~100 GBs of model size
- Irregular accesses with high BW demand (1+ TB/s BW utilization in A100 GPU)

Memory Optimizations (today)

	Training	Inference
ID mapping	Direct-mapped hashing	Direct-mapped with row-wise pruned (~2x reduction)
Precision	fp16 + stochastic rounding (2x reduction) [1]	row-wise int4 quant (8x reduction)[2]
Optimizer	Row-wise sparse AdaGrad (2x)	N/A
Hierarchy	HBM + DRAM [3, 4]	LPDDR (accelerator) + DDR (host)

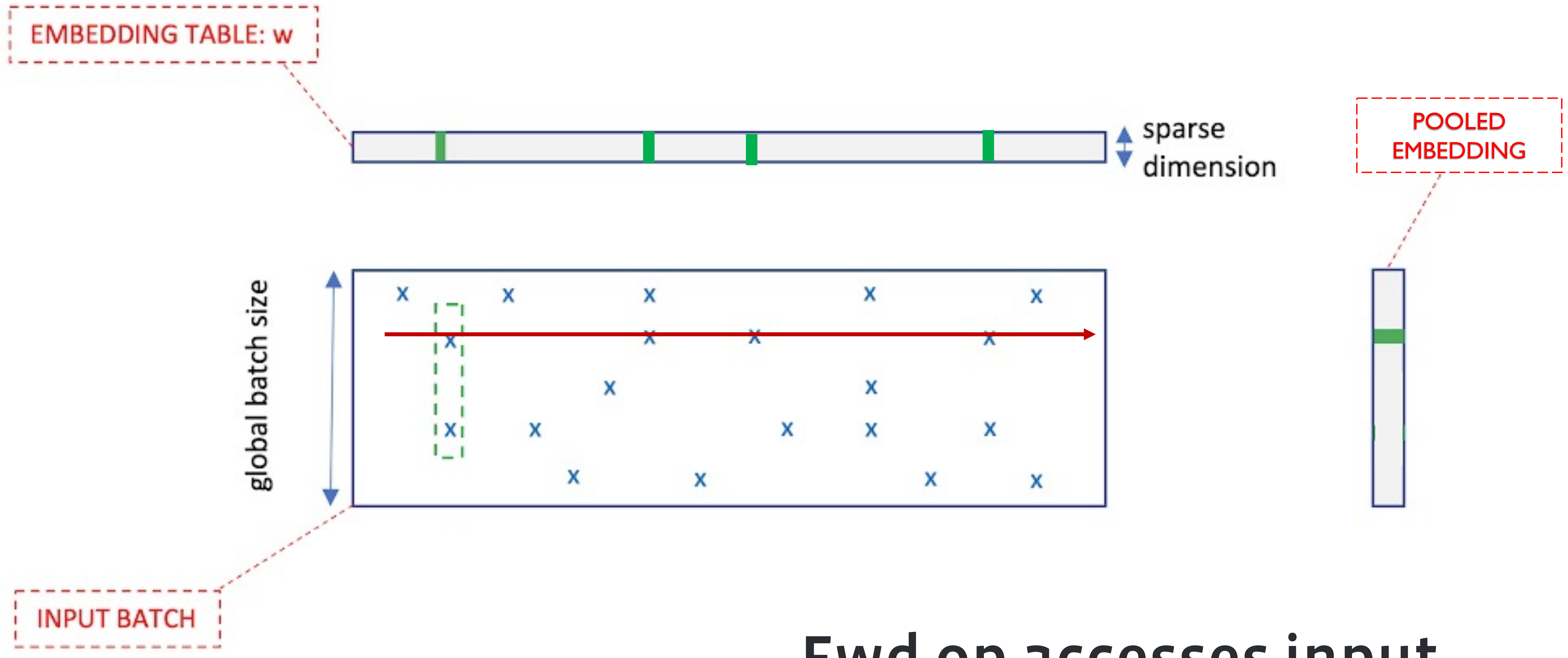
[1] Training with Low-precision Embedding Tables, NeurIPS'18 systems for machine learning workshop
[2] Post-training 4-bit Quantization on Embedding Tables, NeurIPS'19 systems for machine learning workshop
[3] Mixed-Precision Embedding Using a Cache, arxiv
[4] FBGEMM_GPU HBM SW caching code

Memory Optimizations (research)

- Tensor train compression[1]
- HBM + DRAM + SSD hierarchy
- Your idea!

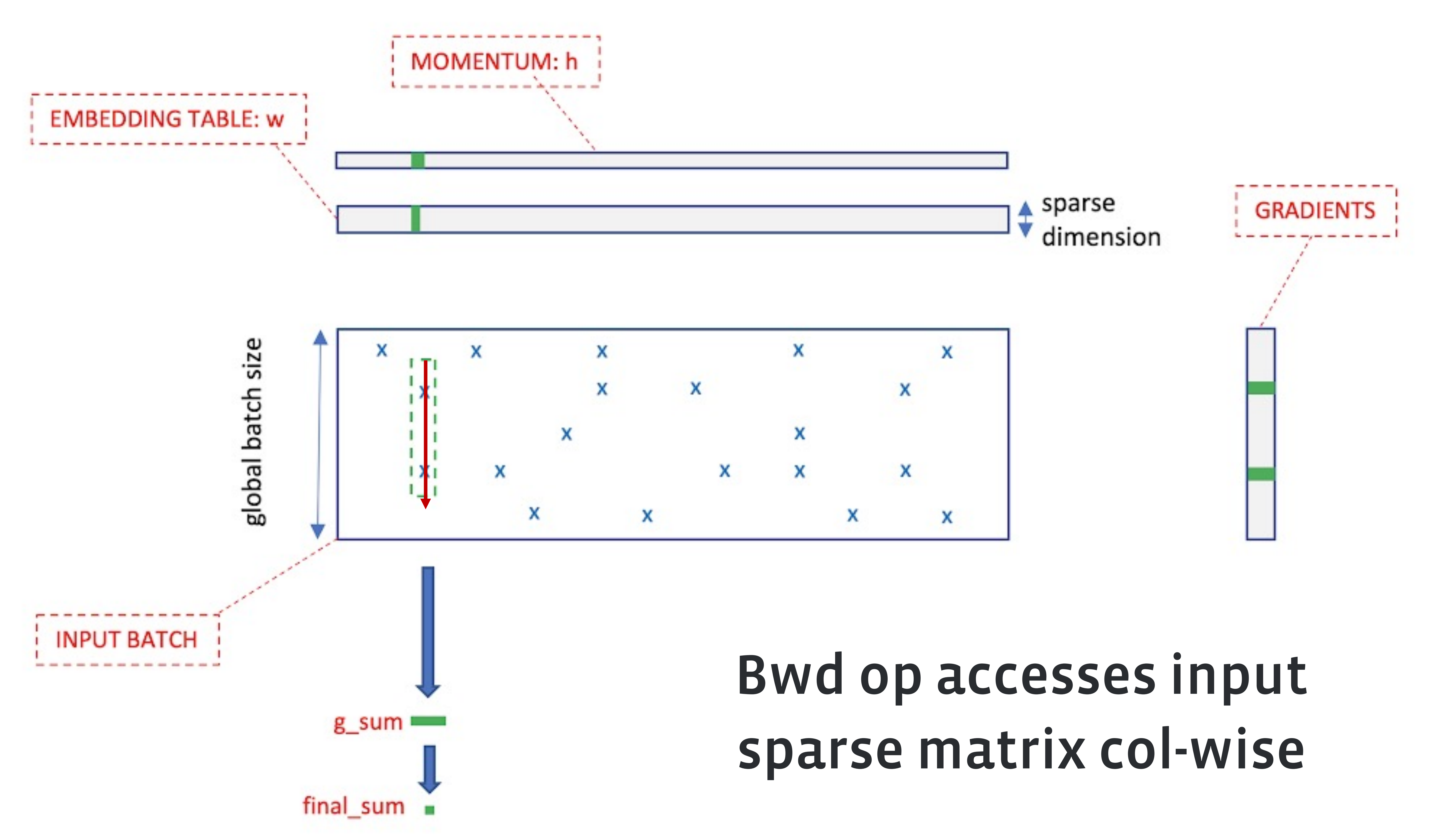
[1] TT-Rec: Tensor Train Compression for Deep Learning Recommendation Models, MLSys 2021

Challenge 2. Exact sparse optimizer



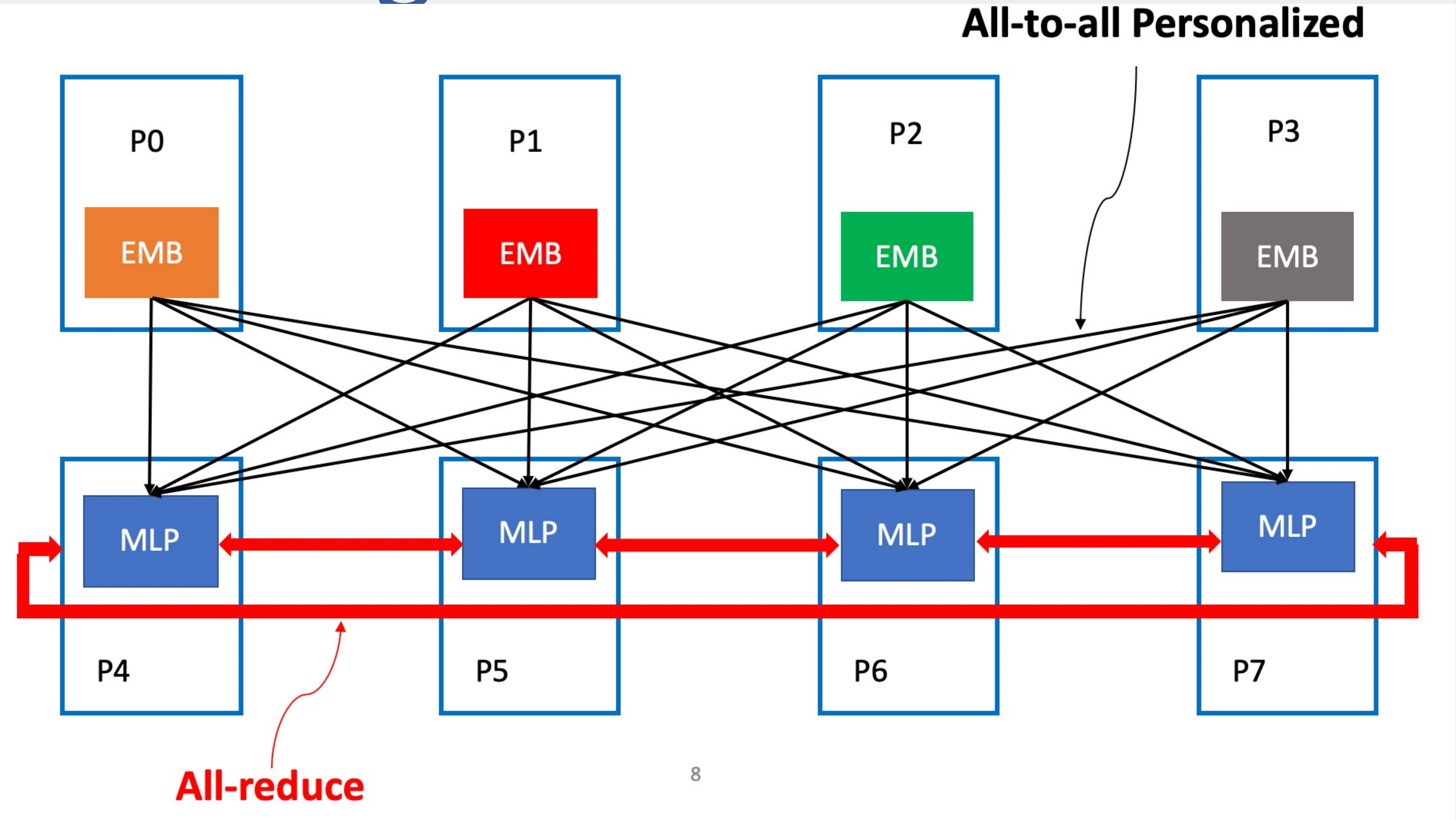
Fwd op accesses input sparse matrix row-wise

Figure credit: Mustafa Ozdal



- Approx. sparse optimizer: update each non-zero individually like Hogwild. Accuracy loss with a large batch size
- Exact sparse optimizer: requires fast parallel sparse transpose

Challenge 3. Communication



Model Parallel: Partition across embedding tables

Data Parallel: Partition across batch dimension

Zion: Facebook Next-Generation Large Memory Training Platform, Smelyanskiy et al., Hotchips'19

Flexible Embedding Table Partitioning

	Note	Index distribution	Fwd	Bwd
Table-wise	Default	all2all	all2all	all2all
Row-wise	Massive tables	bucketization + all2all	reduce-scatter	allgather
Column-wise	To load balance	allgather	all2all	all2all
Data parallel	Small tables			allreduce

- minimize comm + load imbalance
  subject to memory capacity constraints
- Hierarchical: row/column-wise scale-up (e.g., NVLink) + table-wise scale-out (e.g., RoCE, IB)

BW and latency optimizations

- BW
  - Reduced precision (fp16/bf16/int8/int4) communication[1]
- Latency
  - all2all transfers only 10s KBs between a pair of devices

[1] Training Deep Learning Recommendation Model with Quantized Collective Communications: https://dlp-kdd.github.io/assets/pdf/a11-yang.pdf

Summary

- DLRM stresses various aspect of system and embedding operations are in its unique part
- We have done lots of optimizations, but this is just beginning

Our open source projects

- Deep learning recommendation model **reference** implementation: https://github.com/facebookresearch/dlrm
- Facebook GEMM: https://github.com/pytorch/FBGEMM
  - The default reduced precision inference backend of PyTorch
  - **Optimized** CPU/GPU kernels for DLRM

Q&A