AutoShard: Automated Embedding Table Sharding for Recommender Systems

Daochen Zha, Louis Feng, Bhargav Bhushanam, Dhruv Choudhary, Jade Nie, Yuandong Tian, Jay Chae, Yinbin Ma, Arun Kejariwal, Xia Hu

Rice University
Meta Platforms
Background
Embedding Table Sharding Problem

- **Problem Setting**
  - We consider embedding table sharding among GPU devices.
  - We do not consider communication cost.
Key Challenges

• Challenges
  • How to efficiently estimate the cost?
  • How to partition (NP-hard problem).

• Solution
  • Neural cost model
  • Reinforcement learning (RL)
AutoShard

RL Policy

Step Features  Next Step  Sharding Plan

Reward

Micro-Benchmark

Multi-Table Cost Data

Upcoming Table (State)

Shard (Action) 1

Shard (Action) 2

\[
\vdots
\]

Shard (Action) K

Environment

Cost Model

Cost

Cost Model

Element-Wise Sum

1 2 3

Multi-Table Representation
How Does AutoShard Shard?

- **Key Points**
  - Shard sequentially with an LSTM policy.
  - Once trained, it can transfer.
Experiments

• Datasets
  • MetaSyn: https://github.com/facebookresearch/dlrm_datasets
  • MetaProd: around 600 production tables

| Attribute                  | Value                  |
|----------------------------|------------------------|
| Number of Tables           | 856                    |
| Batch Size                 | 65,536                 |
| Max/Min/Min Hash Sizes     | 12,543,670 / 4,107,458 / 1 |
| Max/Min/Min Pooling Factors| 193 / 15 / 0           |

MetaSyn statistics

• Metrics
  • Degree of Balance: min latency / max latency
  • Speedup: max latency speedup over random sharding

• Baselines
  • Lookup-greedy, dim-greedy, size-greedy
Effectiveness

• How is it evaluated?
  • Randomly sample 90 subsets of 80 tables from all the tables as training tasks.
  • Evaluate on another 10 subsets of 80 tables.
  • Shard to 8 GPUs

Performance of AutoShard against baselines
AutoShard Scales to Hundreds of Tables
Efficiency

Training curve on four 2080 Ti GPUs

Inference time with a single CPU core
Summary and Takeaways

- **Embedding table sharding problem**
  - Placing a large number of embedding tables on hundreds of (GPU) devices.
  - Challenges: cost estimation, NP-hardness.

- **Our contributions**
  - AutoShard with neural cost model and RL for sharding.
  - Validated its effectiveness on both open-sourced and production data.

[QR Code for Paper]

[QR Code for Code]