Systems for Machine Learning (CMSC828G)

Optimizing DL Kernels

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

GPU kernels in deep learning

- Important to focus on single GPU and single node performance before looking at scaling/distributed-memory performance
- Requires ensuring that GPU kernels execute as fast as possible
- Two research directions:
  - Systems optimizations - kernel fusion, reducing data movement, etc.
  - ML optimizations - changing the algorithm (for e.g. optimizer used)

Popular kernels

- Dense matrix multiply
- Sparse matrix multiply
  - SpMM: sparse matrix multiplied with a dense matrix
- Dot-product (element-wise) of dense (AB) and sparse matrix (C)
  - SDDMM: Sampled dense-dense matrix multiply
- Attention kernels: dense MM and softmax

Auto-tuning

- Triton's auto-tuning through configs
- torch.compile
- CUDA graphs

torch.compile and CUDA graphs

- torch.compile: capture the computation graph and optimize it
- Major optimizations:
  - Fuse CUDA kernels together
  - Eliminate redundant work
- CUDA Graphs:
  - Eliminates launch overheads by converting lots of small kernel launches into a single graph

torch.compile and CUDA graphs

Feature	`torch.compile`	CUDA Graph
Level	Graph compiler	GPU execution recorder
Purpose	Optimize & fuse ops	Remove CPU launch overhead
Changes kernels?	Yes	No
Fuses operations?	Yes	No
Reduces Python overhead?	Yes	Yes (indirectly)
Requires static shapes?	Helpful	Required
Compile time?	Yes	Minimal capture cost

UNIVERSITY OF
MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu