Siyuan Chen

+1 4128521324 | siyuanc3@andrew.cmu.edu

EDUCATION

Carnegie Mellon University Pittsburgh, USA 09/2023-present
PhD student, Computer Science Department of School of Computer Science
- Advisor: Phillip B. Gibbons, Heather Miller.
- Expected Graduation: 06/2028

Peking University Beijing, China 09/2019–06/2023
Undergraduate student, School of Electronic Engineering and Computer Science
- Major in Computer Science (Turning Class)
- Graduated Summa Cum Laude

PUBLICATION

(*Equal Contribution)

- Siyuan Chen, Zelong Guan, Yudong Liu, Phillip B. Gibbons. Practical offloading for fine-tuning LLM on commodity GPU via learned subspace projectors. Preprint.
- Size Zheng, Siyuan Chen, Siyuan Gao, Liancheng Jia, Guangyu Sun, Runsheng Wang, Yun Liang. "TileFlow: A Framework for Modeling Fusion Dataflow via Tree-based Analysis." 2023 56th IEEE/ACM International Symposium on Microarchitecture (MICRO). IEEE, 2023.
- Siyuan Chen, Pratik Fegade, Tianqi Chen, Phillip B. Gibbons, Todd C. Mowry. "ED-Batch: Efficient Automatic Batching of Dynamic Deep Neural Networks via Finite State Machine." International Conference on Machine Learning (ICML). PMLR, 2023.
- Size Zheng, Siyuan Chen, Yun Liang. "COMB: Memory and Computation Coordinated Mapping of DNNs onto Complex Heterogeneous SoC.", in the proceedings of the Design Automation Conference (DAC-60), July 2023.
- Size Zheng*, Siyuan Chen*, Pedi Song, Renze Chen, Xiuhong Li, Shengen Yan, Dahua Lin, Jingwen Leng, Yun Liang. "Chimera: An Analytical Optimizing Framework for Effective Compute-intensive Operators Fusion", in Proceedings of the 29th international symposium on High Performance Computer Architecture (HPCA-29), February 2023.

Professional Experience

Google New York City, USA 05/2024-08/2024
Student Researcher. System research @ Google (SRG).
Host: Samira Khan.

RESEARCH

Interest: Machine Learning System, Large Language Model, Algorithm design

Memory efficient LLM fine-tuning on commodity hardware via learned subspace projectors 11/2023~ present
Individual research, Carnegie Mellon University
- Fine-tuning LLMs requires significant memory, often exceeding the capacity of a single GPU. A common solution to this memory challenge is offloading compute and data from the GPU to the CPU. However, this approach is hampered by the limited bandwidth of commodity hardware, which constrains communication between the CPU and GPU.
- We designed an offloading framework, DyServe, that enables near-native speed LLM fine-tuning on commodity hardware through learned subspace projectors.
- As a result, our framework can fine-tune a 1.3 billion parameter model on a 4GB laptop GPU and a 7 billion parameter model on a GPU with 24GB memory. Compared to SOTA offloading frameworks, our approach increases fine-tuning throughput by up to 3.33 times and reduces end-to-end fine-tuning time by 33.1%-62.5% when converging to the same accuracy.

TileFlow: A Framework for Modeling Fusion Dataflow via Tree-based Analysis 3/2023-7/2023
Undergraduate Dissertation, Supervised by Prof. Eric Liang, Depart. of EECS, Peking University
- Observed that though with good empirical performance, fusion dataflow for tensor programs is hard to design and compare
- Formulate and automate the design process of fusion dataflow into three stages: Mem-tree design, Tile-tree design, and Loop-tree design.
- Developed a cycle-level simulator to quantize fusion dataflow designs supporting customized software and hardware description.
- Compare and analyze different dataflow designs on different hardware configurations. Design a new fusion dataflow out-performs SOTA dataflow by 1.3x.
- Awarded the top-10 thesis in School of EECS at Peking University.
- Accepted to MICRO23' as co-author.

Optimizations for Dynamic Batching algorithm for Dynamic Neural Networks 07/2022-01/2023
Individual Research, Supervised by Prof. Phillip B. Gibbons, Todd Mowry and Tianqi Chen, Depart. of EECS, CMU
- Observed current technique to exploit batched parallelism for dynamic neural networks is suboptimal.
- Propose FSM-based dynamic batching to find better batching choice. Automate the FSM discovery by RL-based searching algorithm.

- Extended a data structure PQ-Tree to reduce the memory copy by better memory layout;
- Achieved up to 2.4x end2end speedup for DAG-based dynamic neural networks on CPU and GPU compared to state-of-the-art dynamic frameworks, cut down kernel launches by 30~50%, cut down memory transfer amount by 40~50%;
- Accepted to ICML23' as first author.

Mapping Heterogeneous Neural Networks onto Heterogeneous SoC 10/2022-11/2022
Cooperated Research, supervised by Prof. Eric Liang, Depart. of EECS, Peking University
- Observed current mapping framework does not consider the time sharing within one accelerator and treats communication between accelerators equally.
- Proposed a mapping framework to map heterogeneous neural network onto heterogeneous SoC that considers both DNN operators' time sharing, resource sharing, and SoC's heterogeneity in computation and communication.
- Designed and implemented a genetic algorithm to explore the combinational search space.
- Achieved 1.3~1.5x overall latency gain compared to state-of-the-art mappers.
- Accepted to DAC'23 as co-author.

Analytical Model on Kernel Fusion for Compute-Intensive Operator Chains in DNN on CPU and GPU 10/2021-6/2022
Cooperated Research, supervised by Prof. Eric Liang, Depart. of EECS, Peking University
- Observed that compute-intensive operators (GEMM CONV) become memory intensive in DNN workloads.
- Applied aggressive kernel fusion to compute intensive operator chains for better locality and characterize the design space by an analytical model.
- Designed a constant complexity algorithm to solve for the best loop transformation configuration for fused kernels.
- Implemented an auto-scheduler based on TVM and achieve 1.5~2x speedup on CPU compared to vendor libraries and state-of-the-art tensor compilers.
- Accepted to HPCA' 23 as the common first author.

Current Project

Portable GPU-accelerated ML on the edge via WASM 11/2023~ present
Individual research, Carnegie Mellon University
- WebAssembly(WASM) came out as a promising language to run portable code safely in Web/Edge settings.
- Now, WASM cannot drive GPU, making machine learning applications hard to run efficiently through WASM.
- Drove ML applications in WASM by building WASM runtime on top of GPU libraries (CUDA library for NVGPU, WebGPU for the browser)
- Aim to support general GPU programming in WASM.

Skills

Programming Language: Proficient in C++, python;
Framework: Pytorch, Tensorflow, TVM, DyNet.