YUKE WANG

805-259-9421 ⋄ yuke_wang@cs.ucsb.edu

EDUCATION

Ph.D. candidate, Computer Science. GPA: **3.86/4.00** 09/2018 – Now
University of California, Santa Barbara
B.E., Software Engineering. GPA: **3.93/4.00**, Rank: **10/759** 07/2018
University of Electronic Science and Technology of China

SKILLS

Languages: Python, C/C++, CUDA, Linux Shell.
Tools: Pytorch, NVProf, Tensorflow, Scikit-Learn, Vivado HLS, Intel FPGA SDK.

EXPERIENCE HIGHLIGHT

Rich hands-on experience on: 1) designing and optimizing GPU kernels for Graph Neural Networks, such as GCN, GraphSage; 2) developing deep neural networks (**DNNs**), such as ResNet, and VGG.; 3) deploying **machine learning** techniques (random forest, logistic regression, etc.) for **bigdata analysis**; 4) working on **bigdata sets** such as ImageNet.

INTERNSHIP

Research Intern, Alibaba Group DAMO Academy, Sunnyvale, US. 07/2020 – 10/2020

Compiler framework and runtime system for Graph Neural Networks (GNNs) acceleration on GPUs.

RESEARCH & PROJECTS

Optimizing GNN execution flow on GPUs 10/2020 – Now

Explore and implement a set of GNN execution flow (i.e., computation graph) GPU-oriented optimizations tailored for various unseen GNN models generated by GNN-NAS (Neural Architecture Search).

Optimizing CNNs via Sliding-Channel Convolutions 07/2020 – 10/2020

1) Propose and implement the first optimized design for exploring deep separable convolution on CNNs; 2) At the algorithm level, we incorporate a novel sliding-channel convolution (SCC), featured with filter-channel overlapping to balance the accuracy performance and the reduction of computation and memory cost; 3) At the implementation level, we build an optimized GPU-implementation tailored for SCC by leveraging several key techniques, such as the input-centric backward propagation and the channel-cyclic optimization; 4) Integrate the SCC into the existing Pytorch framework as a new type of convolution operator.

Efficient Graph Neural Networks on GPUs 01/2020 – 05/2020

1) Develop a GPU-based GNN acceleration framework. It abstracts and parameterizes GNN execution paradigms (e.g., Sparse Matrix Multiplication, Message Passing), and automatically crafts an optimal execution based on several factors, such as the GNN model architectures and dataset properties; 2) it also incorporates a novel low-cost workload scheduling scheme to exploit the data spatial and temporal locality through graph structural aware thread mapping, which can largely improve the GPU shared memory access efficiency under the large graph and high embedding dimensionality settings.

Dynamic Sparsity-aware Neural Network Inference 11/2019 – 01/2020

1) Develop a light-weighted auxiliary “little” module with random projection and weight quantization for probing Neural Network (NN) layerwise output sparsity to facilitate NN inference acceleration; 2) The proposed ”little” module inference scheme can be easily applied to various types of neural networks, such as CNN, and LSTM. (e.g., on ResNet-18, our proposed method outperforms the state-of-the-art solutions with much higher FLOPs reduction, memory saving and model accuracy).

Performance and Power Efficient Accelerator for GNNs 09/2019 – 11/2019

Improve the aggregation step of GNNs accelerator through on-chip computation reuse; Implement an efficient sparse matrix transformation based on Locality-Sensitive Hashing and Row-Column Reordering to guide PE workload mapping.

Accelerating Distance-related Algorithms on Reconfigurable Hardware 04/2019 – 09/2019

1) Build a compiler framework for accelerating distance-related algorithms (KNN, N-Body Simulation, etc.) on CPU-FPGA platform; 2) Develop a Domain-specific Language (DDSL) and algorithm-hardware co-design to simplify design workflow.

Optimizing CNN Architectures Based on Information Field 02/2019 - 05/2019

Propose a new static analysis method for CNN structure- and kernel- level optimization; The proposed optimization can save 50% FLOPs and 90% parameters on most state-of-the-art CNNs.

FPGA Design Framework for K-means 01/2019 – 04/2019

Build a comprehensive FPGA design framework for K-means acceleration; The proposed design framework combines an algorithmic and a hardware design optimization.

KPynq: A Work-Efficient Triangle-Inequality based K-means on FPGA 10/2018 – 01/2019

Develop a novel design for K-means acceleration on Xilinx Pynq-Z1 FPGA board; KPynq achieves speedup (up to 4.2×) and energy-efficiency (up to 218×) compared with the CPU baseline.

PUBLICATIONS

Yuke Wang, Boyuan Feng, Yufei Ding. ***DSXplore: Optimizing Convolutional Neural Networks via Sliding-Channel Convolutions*** [IPDPS-2021 Submitted]

Yuke Wang, Boyuan Feng, Yufei Ding. ***Optimizing Convolutional Neural Network Architecture via Information Field*** [AAAI-2021 Submitted]

*Boyuan Feng, ***Yuke Wang**, Xu Li, Shu Yang, Xueqiao Peng, Yufei Ding. ***SGQuant: Squeezing the Last Bit on Graph Neural Networks with Specialized Quantization*** [ICTAI-2020 **Accepted**]

Yuke Wang, Boyuan Feng, Gushu Li, Shuangchen Li, Lei Deng, Yuan Xie, Yufei Ding. ***GNNAdvisor: An Efficient Runtime System for GNN Acceleration on GPUs.*** [ASPLOS-2021 Submitted]

Liu Liu, Lei Deng, Zhaodong Chen, **Yuke Wang**, Shuangchen Li, Jingwei Zhang, Yihua Yang, Zhenyu Gu, Yufei Ding, Yuan Xie. ***Boosting Deep Neural Network Efficiency with Dual-Module Inference.*** [ICML-2020 **Accepted**]

Xiaobing Chen, **Yuke Wang**, Xinfeng Xie, Xing Hu, Abanti Basak, Ling Liang, Mingyu Yan, Lei Deng, Yufei Ding, Zidong Du, Yunji Chen, Yuan Xie. ***Rubik: A Hierarchical Architecture for Efficient Graph Learning.*** [TCAD-2020 Submitted]

Yuke Wang, Boyuan Feng, Gushu Li, Lei Deng, Yuan Xie, Yufei Ding. ***AccD: A Compiler-based Framework for Accelerating Distance-related Algorithms on CPU-FPGA Platforms.*** [ASPLOS-2021 Submitted]

Yuke Wang, Boyuan Feng, Gushu Li, Georgios Tzimpragos, Lei Deng, Yuan Xie, Yufei Ding. ***TiAcc: An Efficient Triangle-Inequality based K-means Hardware Accelerator Design on FPGA***. [CCGrid-2021 Submitted]

Yuke Wang, Zhaorui Zeng, Boyuan Feng, Lei Deng, and Yufei Ding. ***KPynq: A Work-Efficient Triangle-Inequality based K-means on FPGA***. [FCCM-2019 **Accepted**].

HONORS & AWARDS

2020 Summer GSR recipient in CS Department of UCSB	06/2020
2019 Summer GSR recipient in CS Department of UCSB	06/2019
Outstanding Graduates Award of UESTC	10/2017
First-class People's Scholarship (2/20 in the Elite Program)	10/2017
Interdisciplinary Contest In Modeling (ICM) [Honorable Mention]	04/2017
Suzhou Industrial Zone Scholarship (2/20 in the Elite Program)	04/2017
International Software Testing Qualifications Board (Certified Tester) [Foundational Level]	10/2016
First-class People's Scholarship (4/116)	04/2016

EXPERIENCE

Teaching Assistant of CS160 (Translation of Programming Languages)	Fall. 2019
Teaching Assistant of CS8 (Python Programming Language)	Summer. 2019
Teaching Assistant of CS16 (C++ Programming Language)	Winter. 2019

PROFESSIONAL ACTIVITIES

Program Committee:
AAAI-2021 (Top artificial intelligence conference)

Reviewed Papers:
ASPLOS-2020, ASPLOS-2019 (Top computer system & architecture conference)
PPOPP-2020 (Top parallel programming & computer system conference)
OOPSLA-2019 (Top programming language conference)

Presentations
Invited talk at Computer Systems WIPS: KPynq: A Work-Efficient Triangle-Inequality based K-means on FPGA. Feb. 2019