Congrui (Jerry) Yin

✉ yin00486@umn.edu | github.com/JerryYin777 | jerrysys.top | Google Scholar | Congrui Yin

Education

University of Minnesota Twins Cities 2024/01 - 2025/06 (Expected)
Bachelor of Arts in Computer Science, Minor in Statistics Minneapolis, MN, USA

Nanchang University 2021/09 - 2023/12
Candidate for B. Eng in Artificial Intelligence Nanchang, Jiangxi, China

- Enterprise Special Scholarship, 2023. **(Only 30 in School)**
- School Special Academic Scholarship, 2023. **(1%)**
- School First-Class Academic Scholarship, 2022. **(8%)**

Research Interests

I am broadly interested in the intersection between natural language processing and efficient machine learning system (Mainly based on GPU). My previous work was in building **efficient computation systems for NLP training & inference and supercomputer scientific applications.**

Publications

- **F-PABEE: Flexible-Patience-Based Early Exiting For Single-Label and Multi-Label Text Classification Tasks**. X. Gao, W. Zhu, J. Gao and **C. Yin**. (2023). *2023 IEEE International Conference on Acoustics, Speech and Signal Processing* (**ICASSP 2023**). [Paper]
- **Multi-scale and multi-task learning for human audio forensics based on convolutional networks**. **C. Yin**. (2023). *International Conference on Image, Signal Processing, and Pattern Recognition* (**ISPP 2023**). [Paper]

Research Experience

TsinghuaNLP 2023/07 - 2023/09
LLM System Algorithm Research Assistant, advised by Prof. Zhiyuan Liu Beijing, China

- My research focuses on distributed AI training systems, specifically addressing training methods for neural networks at a **scale of trillions of parameters**. As the size of AI models increases, the time complexity of Transformer architecture models escalates, prompting the need to explore more effective architectures for training. In this context, I endeavor to train models using the RetNet architecture within the **BMTrain** distributed training framework that I have developed.

GOOD Lab, Nanchang University 2021/12 - 2023/12
High-Performance System Algorithm Research Assistant, advised by Prof. Zichen Xu Nanchang, Jiangxi, China

- I have been engaged in long-term research on Serverless Computing (Function as a Service) at GOOD Lab, to simplify AI deployment, enhance efficiency, and reduce costs through cloud computing. As part of this research, I undertook stress experiments on AI services utilizing Kubernetes and Docker on supercomputers.

Work Experience

Zhihu & ModelBest Ltd. 2023/07 - 2023/09
Algorithm Intern Beijing, China

- In collaboration with TsinghuaNLP, I simultaneously worked on Zhihu (Chinese Quora) and ModelBest Ltd. During this partnership, I utilized the highest-quality Chinese corpus available on Zhihu for training an 80b multimodal large model. I also played a significant role in creating the distributed LLM training framework **BMTrain**, which successfully addressed communication bottlenecks during the training of large language models. When compared to DeepSpeed, this tool achieved a **1.6x increase in throughput** for training Zhihu LLM.

Open-Source Contributions

OpenBMB Community 2023/07 - 2023/09

- **Contributor of** BMTrain **(★454)**. BMTrain is an efficient large model training toolkit that can be used to train large models with tens of billions of parameters. It can train models in a distributed manner while keeping the code as simple as stand-alone training, used by ChatGLM-6b, CPM-20b, Luca-80b LLMs.

- I implemented the Zero-offload method based on Triton and CUDA within BMTrain. This allows memory occupancy to replace GPU memory usage, reducing the computational load for training large language models. The successful implementation of distributed training was achieved on a cluster of 512 A100 GPUs.
- Additionally, I added support for bf16 and fp8 data types specifically for the A100 and H100 architectures and implemented optimizations for the corresponding Adam Optimizer and learning rate schedule.

CGCL-Codes 2023/03 - 2023/06

- **Contributor of** NaturalCC **(★225)**. NaturalCC is a sequence modeling toolkit designed to bridge the gap between programming and natural languages through advanced machine learning techniques. It allows researchers and developers to train custom models for a variety of software engineering tasks, e.g., code generation, code completion, code summarization, code retrieval, code clone detection, and type inference.
- I enhanced the Transformer architecture based on AST syntax tree principle, making the construction of large-scale code language models more abstract at a lower level.
- Additionally, I extended its compatibility from only using Fairseq to supporting Transformers, including popular large code models from Hugging Face such as Codellama, CodeT5, CodeGen, and StarCoder.

Personal Projects 130 followers, 800+ Stars JerryYin777

- **NanoGPT-Pytorch2.0-Implementation** (★52) NanoGPT Implementation based on Pytorch 2.0 (when Pytorch 2.0 released soon on Mar. 2023), faster and simpler, a good tutorial learning GPT.
- **Intelligent-Creator** (★2) I implemented the Intelligent Creation Platform Creator, which comprises a front-end and back-end separation architecture software for generating titles and summaries based on Chinese news text using the GPT-2 model. This implementation predates the ChatGPT era.
- **Cr's Research Toolchain** (★52) Share my research toolchain

Selected Awards

I was the leader of Nanchang University Student Cluster Competition Team (**Team NCUSCC**), participating in world's largest supercomputer competition **ASC22** and **SC23.**

ASC22 (5-people Group) 2022/01 - 2022/03

- Ranking 23/500+, Second Prize in World Class
- **Yuan-LLM** (★6) I employed parallel frameworks and methods such as Megatron-LM, ZeRO, and DeepSpeed for distributed training of the largest Chinese language model (2022.2) YUAN-18B with 8 V100 GPUs in two servers.

SC23 (6-people Group) 2023/03 - 2023/11

- Ranking 7/15 (Among MIT, Brown, Tsinghua U, Gatech, Peking U, etc.)
- I utilized the SLURM management tool to successfully execute HPL benchmark tests on **300 servers** using parallel methods such as MPI, OpenMP.

Technical Skills

- **Languages:** Python, C/C++, CUDA, Go, Rust, Shell, LaTeX
- **Frameworks and Tools:** Pytorch, JAX, Triton, Docker, Kubernetes, MPI, OpenMP, AWS, Sklearn, Numpy, RISC-V
- **AI:** Natural language Processing (llama-2, ChatGLM-3, CPM-Bee) | MLSys (Flash attention & ZeRO Series) | Computer Vision (YOLO Series, OpenCV, Simple Ray Tracing) | Multimodal Pretrained Model (BLIP-2, LLAVA)

Last Updated on January 21, 2024