Congrui (Jerry) Yin

yin00486@umn.edu | JerryYin777 | jerrysys.top | Google Scholar

Education

University of Minnesota Twin Cities 2023/12 - 2024/12 (Expected)
B.A. in Computer Science GPA:3.85/4.0 Minneapolis, MN

Nanchang University (Transfer Out) 2021/09 - 2023/12
B. Eng. in Artificial Intelligence Nanchang, China

- School Special Scholarship, 2023. **(1/30)** Special Academic Scholarship, 2022 & 2023. **(1%)**

Research Interests

I have experience in NLP and computer systems fields (both architecture and high performance machine learning systems). My current passion revolves around building **EFFICIENT** system solutions to AGI and LLM (VLM) for **RELIABLE** Hardware Design, this includes:

- Machine Learning System
  - Training: Design more effective training system and algorithms, examples include **BMTrain** (★541).
  - Quantization
  - Long context inference: example includes **Cross-Layer-Attention** .
- LLM (VLM) for **RELIABLE** Hardware Design
  - Synthesise my pretraining and finetuning common knowledge of CodeLLM, exploring the boundary capabilities of LLM/VLM for hardware design (e.g. pretrain/finetune a VerilogLLM).

Publications

- **IAPT: Instance-Aware Prompt Turing for Large Language Models**. Y. Ni, **C. Yin**, A. Tian, X. Wang, G. Xie. (2024). *The 62nd Annual Meeting of the Association for Computational Linguistics* (**ACL 2024**).
- **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]

Research Experience

TsinghuaNLP 2023/07 - 2023/09
Research Assistant, supervised 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**. I played a role in creating the distributed LLM training framework **BMTrain**, which successfully addressed communication bottlenecks while training large language models.
  - 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. Distributed training was successfully implemented on a cluster of 128 A100 GPUs.
  - 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 scheduler.
  - BMTrain achieves a **1.4x increase in throughput** for training a specific LLM compared to MegatronLM.

Industry Experience

SenseTime Research 2024/05 - 2024/08
Algorithm Research Intern Beijing, China

- Develop **CodeLLM Raccoon (Copilot-like Coding Assistant)**
  - Finetune a multimodal automatic error correction model for code and tabular data.

01.AI 2024/02 - 2024/04
Algorithm Research Intern Remote

- Develop **Yi-Large Large Language Model (China's top-ranked model in LMSys Leaderboard) ML Infra**

Open-Source Contributions

CGCL-Codes 2023/03 - 2023/06

- **Contributor of** NaturalCC **(★253)**. 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 the AST syntax tree principle**, making the construction of large-scale code language models more abstract at a lower level.
- I extended its compatibility from only using Fairseq to supporting Huggingface Transformers, including popular large code models from HuggingFace such as Codellama, CodeT5, CodeGen, and StarCoder.

Personal Projects 260+ followers, 800+ Stars JerryYin777

- **PaperHelper** (★9) With the effects of RAG Fusion RAG Finetune (using GPT-4 API on the 52,000 MLArxivPapers and ArxivQA dataset as the backend), it can effectively reduce hallucinations and enhance retrieval relevance by providing references ranked by relevance and using structural relationships to represent the extracted information.
- **Intelligent-Creator** (★3) 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.
- **RISC-V Processor Core** (★10) I Implement a three-stage pipelined, single-core, 32-bit, small RISC-V processor core using Verilog.

Technical Skills

- **Programming Languages:** Python, C/C++, CUDA, Verilog (Chisel), Shell.
- **Frameworks and Tools:** Pytorch, Triton, Huggingface, Vivado.
- **MLSys:** FlashAttention-1,2, QLoRA, DoRA, vLLM, etc.

Misc

- **Language:** English (Fluent, TOEFL 112/120) | Mandarin (Native)
- **Writing:** I love sharing my knowledge about MLSys and NLP on 知乎 JerryYin777 (in Mandarin) with **3.8k followers**.

Last Updated on August 9, 2024