Yan (Melody) Zhao

Mobile: (206) 843 7025 | Email: zhao.y4@northeastern.edu | LinkedIn | GitHub | Hugging Face | **Seattle WA**

EDUCATION

Northeastern University **Seattle,Washington**
Master of Science in Computer Engineering **(GPA:4.0/4.0)** **Jan 2025 - May 2027**
Coursework: Machine Learning, Data Structure and Algorithms, Object-Oriented Programming, Web Development, High Performance Computing for AI, Natural Language Processing(NLP), Classical Machine Learning and Data Science

SKILLS

Programming Languages: Python, Java, JavaScript, TypeScript, SQL, HTML/CSS
Cloud Platforms: AWS, Google Cloud (Vertex AI), Docker, Azure
Databases & Storage: PostgreSQL, MongoDB, Vector Databases (ChromaDB)
Software Engineering: Object-Oriented Programming, Data Structures & Algorithms, System Design, Microservices Architecture, RESTful APIs, Distributed Systems
Full Stack Framework: React, Node.js, Angular, Spring Boot, Flask, Webpack, WebSocket, Streamlit
AI/ML Frameworks: PyTorch, TensorFlow, Hugging Face Transformers, LangChain, Large Language Models (LLMs), RAG Architecture, Model Fine-tuning using Unsloth (LoRA/PEFT), NLP, MLOps, VLLM, SGLang
DevOps & Version Control: Git/GitHub, CI/CD Pipelines, Docker Compose, Linux

WORK EXPERIENCE

Human Ageing Genomic Resources | AI Engineer Intern | Seattle WA May 2025 - Sep 2025

- Developed an end-to-end LLM-powered biomedical chatbot with **RAG** using Python, **ChromaDB**, and **Streamlit frontend**; implemented agent orchestration framework (**Agno**) to coordinate retrieval, reasoning, and synthesis agents for intelligent query routing and multi-step biomedical question-answering, reducing researcher query time by 60% through real-time streaming responses, source citation tracking, and conversation history management.
- Engineered **data preprocessing pipelines** to clean and optimize institution-provided biomedical datasets; unified diverse **data formats** (JSON, CSV, XML), filtered noise and resolved inconsistencies, applied text normalization and tokenization techniques, and transformed unstructured content into structured representations for high-quality vector embeddings in RAG retrieval.
- **Fine-tuned Gemma** LLM on **Google Cloud TPU** using **PyTorch** with **Unsloth** framework and **LoRA** (Low-Rank Adaptation) for parameter-efficient training, reducing trainable parameters to <1% of the full model while maintaining strong performance on biomedical question-answering tasks; optimized **hyperparameters** including learning rate, LoRA rank, and target modules for domain adaptation.
- Deployed fine-tuned model using **vLLM** inference engine on **Lightning AI GPU** infrastructure.Built real-time data pipelines integrating **ChromaDB vector database** and **SQL query engines**, enabling dynamic retrieval from both **structured** (clinical records, gene databases) and **unstructured** (research literature) biomedical datasets with sub-second query response times.

Tianjin Motor Dies Co.,Ltd. | Technical Solutions Engineer & Project Manager | Tianjin China Sep 2008 - Dec 2018

- Led multiple **$10M+** automotive manufacturing projects with **50+** engineers across global teams for Fortune **500** clients (Tesla, Ford, GM, Land Rover, Fiat) . Improved delivery efficiency by **30%** via data-driven project workflows

Teaching Assistant | Northeastern University| Seattle WA Sep 2025 - Present

- Instructed **Python** programming to **30+** students weekly for INFO5002:Introduction to Python Programming

TECHNICAL PROJECTS

Deep Learning Fundamentals & Autograd Engine
Micrograd Extension | GitHub Inspired by Andrej Karpathy: Self-studied automatic differentiation by implementing Micrograd from scratch to understand the computational backbone of **deep learning frameworks** like **PyTorch**; built the Value class with operator overloading, implemented topological sort for reverse-mode **backpropagation** through dynamically constructed directed acyclic graphs (**DAG**), added custom operations (**ReLU**, **tanh**, **exponentiation**), and developed a complete MLP training loop with gradient descent and loss visualization—gaining deep understanding of how production frameworks handle automatic differentiation, gradient computation, and memory management across arbitrary neural architectures.

High-Performance Distributed ML: Parallelism Techniques

- Implemented **data parallelism MNIST training** with JAX's shardmap sharding API and **contributed to OSS JAX | GitHub**
- Deep-dived into Microsoft's **ZeRO paper** and built toy implementations of ZeRO Stage 1 (optimizer state sharding), Stage 2 (gradient sharding), and Stage 3 (parameter sharding) using PyTorch to understand memory-efficient model parallelism; validated memory savings and communication patterns across multi-GPU setups, gaining practical understanding of how frameworks like DeepSpeed enable training of billion-parameter models. | **GitHub**
- Walked through **VLLM** code base and summarized understanding of inference perfomance topics, such as KV cache management, Paged Attention, batching techniques such as continous batching.

Multi-Channel E-commerce Platform | GitHub

- Architected and implemented **object-oriented Java** application with layered architecture, featuring gui and analytics dashboard

Open source contributor | GitHub Contributed to **JAX**, **HuggingFace Transformers**, and **LangChain**