Hui Xu

Stony Brook, NY, United States of America
☎ (516)457-4066 ✉ huixucom@gmail.com in linkedin.com/in/huihsuxu/ 🌐 huixu11.github.io

EDUCATION

Masters in Engineering Artificial Intelligence - GPA: 3.67/4 **Aug 2024 - Dec 2025**
State University of New York - Stony Brook *NY, USA*
Coursework: **Distributed System**, **Natural Language Processing**, Deep Learning Algorithms and Software
MAQuA: Adaptive Question-Asking for Multidimensional Mental Health Screening using Item Response Theory
Accepted by EACL*(view paper)*

Masters in Computer Application Technology - GPA: 4/4 **Sep 2013 - June 2018**
Beijing Forestry University *Beijing, China*

Bachelor in Information Management and Information Systems - GPA: 3.9/4 **Sep 2009 - June 2013**
Beijing Forestry University *Beijing, China*

WORK EXPERIENCE

Mastercard | *Django,* ***React, Python,*** *Redux,* ***Typescript,*** *PostgreSQL, SQLite, Ray* **Nov 2021 - Jul 2024**
Software Engineer II *Beijing, China*

- Built a **full-stack** business analytics platform ("Test & Learn") from the ground up for Chinese banks, enabling local deployment and data compliance, using Django (backend), **React**/Redux (frontend), and PostgreSQL/SQLite.
- Proposed and led migration from an in-house multiprocessing framework to Ray Core, enabling distributed execution across clusters and containerized environments with minimal code changes.
- Developed a high-performance outlier detection algorithm using statsmodels leave-one-out statistics; improved runtime **from 9 hours to 10 minutes** through selective computation and vectorization.
- Designed and implemented a script generator for large-scale dummy datasets with O(1) space complexity, significantly improving data-simulation efficiency for testing.
- Led the design and delivery of a Driver Summary module showing driver significance and visual summaries via React, Redux hashmaps, and Recharts; **collaborated with PMs and tech leads to refine product requirements and architecture**.
- Architected and implemented a Metric Uploader feature capable of processing **400MB+ CSV files in under one minute**, with row-level validation using a fully vectorized algorithm and comprehensive unit testing coverage.
- Designed data structures and algorithms for dynamic result grid manipulation using react-beautiful-dnd and Redux, enabling flexible drag-and-drop analytics views.
- Demonstrated strong leadership and ownership—regularly identified technical roadblocks, **coordinated across engineering and product teams**, and drove feature completion under tight timelines.

Dazhangfang (Chinese Intuit) | ***SQL,*** *Redis, APScheduler, OCR, Flask,* ***Google Cloud*** **July 2018 - Oct 2021**
Python Engineer *Beijing, China*

- Maintained and enhanced the company's invoice and bank form recognition system and finance/taxation APIs, enabling users to upload receipts for automatic recognition, classification, and accounting.
- Managed and deployed a **large-scale recognition platform (100,000 lines of code, 10 servers)** integrating the recognition engine, invoice verification service, and web service for recognized results.
- **Optimized database queries and indexing**, improving the Invoice Recognition Web Service performance by 99.99%, dramatically reducing response latency.
- Automated manual invoice verification, achieving a 90% reduction in human intervention using edit-distance algorithms for text matching and validation.
- Implemented asynchronous task scheduling and message delivery using APScheduler and Redis as a message queue broker, increasing throughput and reliability.
- Designed caching mechanisms for recognition results and optimized the end-to-end OCR pipeline, significantly reducing compute cost and improving system scalability.

PROJECTS

Transformer Language Model from Scratch (CS336) | *Python,* ***PyTorch,*** *NLP, BPE,* ***Transformer,*** *CUDA*

- https://github.com/helperfunc/assignment1-basics
- Engineered a Transformer LM from scratch with BPE tokenizer, RMSNorm, RoPE, multi-head attention, and SwiGLU layers.
- Built and tested Linear, Embedding, and Attention modules ensuring stable gradients and efficient backprop.
- Trained on TinyStories/OpenWebText using multiprocessing pre-tokenization and custom AdamW optimizer.
- Achieved < 2 min BPE training (10 K vocab) and fluent text generation with competitive perplexity.

Jane Street GPU Mode Hackathon 10th Place | *Python, PyTorch*

- https://github.com/helperfunc/hackathon
- Optimized latency and accuracy of the code using **dynamic batching strategies**.
- Achieved About 71.1ms latency and 0.77 accuracy, earn $127/s.