WENHE LI

New York, NY |6468756387| wl692@cornell.edu |Github: https://github.com/WenheLI | Portfolio: https://wenhe.li

EDUCATION

Cornell Tech, New York, NY — Aug 2021-May 2023
Jacobs Technion-Cornell Dual Master of Science Degrees – Connective Media Concentration | GPA: 4.0

New York University, Shanghai, China & New York, NY — Aug 2016-May 2020
Bachelor of Science in Computer Science with Honors | Major GPA: 3.8
Dean's List & University Honors Scholar & Computer Science Honors Degree

TECHNICALS

Coding Language: JavaScript/TypeScript, C/C++, Python, Java, Kotlin, Dart, Golang, C#, SQL, Rust
Software Engineering Tools: React, Express, TailwindCSS, MongoDB, AWS Lambda/Neptune/DynamoDB, Spring Boot
Deep Learning Tools: TensorFlow.js, TensorFlow, XNNPACK, TVM, PyTorch

EXPERIENCE

Microsoft, Software Development Intern, Redmond, Seattle — May 2022-Present

- Implemented ***OpenTelemetry*** Metadata Exporters in ***Golang*** and ***C#***. Enable the ability to author ***SLO*** automatically.
- Developed ***RESTFul API*** for storing and retrieving metadata from ***Azure Purview*** and ***Azure DataLake***.
- Contributed to the ***OpenTelemetry*** Open-Source Community by adding new features and drafting specifications.

Google Summer of Code 2021 - Chromium, Student Developer, Remote — May 2021-Aug 2021

- Implemented Android NNAPI (C++) with XNNPACK for ChromeOS. ***Achieved a 30% improvement*** in CPU inference time.
- Implemented dynamic load for different backend drivers with the ***shared library***; Allowed switching drivers in execution time.

Alibaba, Software Development Engineer, Hangzhou, China — Jun 2020-May 2021

- Developed promotion website; Added on-device user-intention detection; ***Achieved 3% improvement*** in coupon utilization.
- Core developer of Pipcook, an open-source Deep Learning (DL) framework for JavaScript and large-scale training.
- Reused the DL ecosystem for the ***Web*** by using ***TVM*** & ***Emscripten*** to compile DL mode and C++ code into WebAssembly.
- Implemented ***Merchant Feeds*** and ***on-device re-rank*** model for it. ***Achieved a 10% increase*** in Click Through Rate (CTR).
- Combined ***user-intention detection*** with layout recommendation; Achieved customized User Interface and ***10% CTR increase***.
- ***Embedded*** Node.js into ***Distributed System*** and ***implemented*** Database I/O; Allowed execution of large-scale tasks in JavaScript.

Alibaba, Frontend Development Internship, Hangzhou, China — Jun 2019- Oct 2019

- Developed ***GBDT runtime*** in JavaScript; Used it to detect leaving patterns in websites; **Achieved a 5% decrease** in bounce rate.
- Implemented text analysis for code and constructed a pipeline for training and deploying the text analysis model.
- Integrated above model to ***UI auto-generating*** solution; Got **69% accuracy** in practice; Increased developing efficiency by 30%.
- Simplified training/deploying models and sample collecting workflow by developing a Web App(React) and Backend(Koa).

Google Summer of Code 2019 - TensorFlow, Student Developer, Remote — May 2019- Sep 2019

- Produced **automation test** solutions for web workers based on Karma.
- Added web worker support for browser and Node.js; Made training and inference over JavaScript in **multithread** possible.

SELECTED PROJECTS

ClusterNetworks, (TypeScript, C++, Emscripten, Python) — Spring 2022
A high-performance 3D multi-omics(medical data) visualization tool (collaborated with Mount Sinai).

- Exported a C++ Graph library to WebAssembly; Offers an efficient solution to operate graphs on browsers.
- Utilized Three.JS to build up a rich interaction 3D Network and Cluster visualization.
- Conducted User Research and polished the user experience while using this visualization tool.

MiniTorch, (Python, Cuda) — Fall 2021
A minimal viable Deep Learning Training framework inspired by Pytorch.

- Implemented essential operators in CPU and CUDA.
- Implemented Tensor Memory Management and Auto-Differentiation.

Hyper.media, (Kotlin, Node.js, Python) — Spring 2020
A live-streaming Application aims to achieve extremely low latency and experiment with its interactions.

- Used **Kotlin** to develop an Android Application with live streaming and AR (ARCore) functionality.
- Reduced bandwidth with on-device Deep Learning Upscaling Model; Achieved **half** of the original bandwidth requirement.
- Benchmarked live streaming latency among different protocols; Achieved **sub-second latency**.

COMMUNITY SERVICE

Google Code-in 2019, Student Mentor, Remote — Nov 2019-Jan 2020

- Designed tasks for annual programming competitions to be solved by high school students.
- Mentored students wanting to contribute to TensorFlow.js in the Google Code-in contest.

CERTIFICATION & PUBLICATIONS

"StoryDroid: Automated Generation of Storyboard for Android Apps", International Conference on Software Engineering, 2019
Machine Learning Google Developer Expert