Daniel Song

daniel.song@uwaterloo.ca
github.com/danwsong
linkedin.com/in/danwsong

Skills

Languages

JavaScript	TypeScript
Python	HTML / CSS
C++	SQL
Go	Swift

Frameworks / Libraries

React	Node.js
React Native	PyTorch
Redux	TensorFlow

Tools

Git	Postgres
Bash	Redis
Linux	Amazon S3
Docker	Amazon EC2

Achievements

2019 USACO Gold Division

2019 ECOO Programming Contest Provincials Finalist

2018 Urban Hacks Finalist

2019 Reach for the Top National Champion

Education

University of Waterloo

Sep 2019 — Apr 2024

Software Engineering (BSE)

Term GPA: 92%
Cumulative GPA: 90%

Interests

Basketball

Analog photography

Retro game development

Music production

Language learning

Experience

Software Engineer — Mentum Group

Jan 2021 — Apr 2021

Engineered redesigns of **React**- and **React Native**-based frontend apps, which serve **11 000+ orders** per week and are used by **more than 80 restaurants** across Ontario

Migrated 80% of the existing frontend codebase from **JavaScript** to **TypeScript** and introduced ESLint and code formatting configurations, improving maintainability and reducing warnings and errors **by more than 95%**

Implemented support for food delivery by integrating external APIs from **DoorDash** and **Postmates** into an existing **Go** backend service

Improved frontend test coverage to **65%** by contributing to **more than 50** end-to-end **Cypress** tests

Machine Learning Engineer — Tealbook

May 2020 — Aug 2020

Developed a **natural language processing**-based web crawling system in **Python** to extract information from company websites using **PyTorch**

Deployed the system on a distributed cluster of **Compute Engine VMs** using **Docker**, extracting 700 000+ datapoints from more than 4 000 000 websites with a **95% accuracy rate**

Expanded an existing **Python** web crawling system to extract information from company websites in an **additional 13 countries** across Asia and Europe

Projects

Browser Game Boy Emulator — gemuboi-js

Developed a browser-based Game Boy emulator in **JavaScript**, with accurate audio support, persistent game save storage support, and support for running Game Boy Color games

Tested on **90+** Game Boy and Game Boy Color games, **more than 98%** of which run without any accuracy issues

Checkers Engine — checkers-engine

Designed and implemented a neural network-based engine in **C**, allowing pre-trained neural networks to be used in place of traditional evaluation functions

Trained a neural network for the engine using **TensorFlow** on a dataset of 150 000+ board positions from 4 000 checkers games

Optimized using techniques including alpha-beta pruning and bitboards, allowing the engine to think **up to 15 moves ahead within 10 seconds**

Arduino Wireless API — pi-arduino-interface

Configured a **Python Flask** server to accept requests over HTTP, translate them into a lightweight command-based protocol, and transmit them to an **Arduino** over USB

Built programs in **C** to decode incoming requests and output data to pins on the **Arduino** accordingly