Alan Yuan

☎ +1-647-918-8482
✉ alan.yuan.jly@gmail.com
in /JaLnYn
/JaLnYn

Experience

Research Assistant in Orbit project | PAIR Lab — Python

September, 2021-Present

- Building a rigid-motion simulator as well as implementing R.L algorithms.

Software Engineer - Intern | Intel — C++, Python, Bash

May 2021 - present

- Work in the FPGA synthesis team, working towards and debugging new elaboration flow.
- Automated mass file changes and testing using Python and Bash scripts cohesivly.
- Random test-case generation for automated compilation testing.

Software Developer | Centivizer — Node.JS: SimplePeer, Socket.io, Axios

April 2020 - Dec 2020

- Fully designed and built full-stack application using NodeJS and SimplePeer to connect users via Video
- Used Socket.io to communicate between the client and backend for the video feature.
- Integrated the video feature with the their user database through a RESTful API using the Axios library.

Education

University of Toronto — Sep, 2018 – Apr, 2023

Honours B.S. in Computer Science — *cGPA: 3.83/4.0*

Relevant Coursework: Data Structures and Algorithms (92), Operating Systems (91), Parallel Programming (94), NNs and Deep Learning (96), Intro to AI (97), Introduction to M.L (95)

Side Projects

CFR Minimization (Kuhn poker, Tic-tac-toe and Coup) — Python, PyTorch

May, 2021-Present — *github.com/JaLnYn/pokerbot*

- Developed a general framework to train R.L. agents using **CFR, CFR+ and MCCFR** to find the nash equilibrium.
- Implemented each of the algorithms to play tic-tac-toe.
- Plan to use the algorithms to solve Kuhn poker and Coup.

CSC311 (Intro to M.L) Course Competition — Python, numpy

Nov, 2020-Dec, 2020 — -

- Achieved the **5th highest score** in the competition and a 99% on the project write-up.
- Chose and implemented a Matrix Factorization algorithm to recommend a selection of movies to users.
- Improved on the SGD training process by adding weight regularization and biases based on paper on a different application.
- Used ensembles to decrease variance ensuring the private score will be similar to that of the validation set.

Tron UDP multiplayer — C++, ncurses

Sep, 2019-Dec, 2019 — *github.com/JaLnYn/Tron*

- Created a four player game for **local networks** using the UDP network protocol and C++
- Forked timer from the server to ensure the game runs on time
- Utilize epoll for both client and server to monitor the socket as well as the timer (server) and stdin (client)

Skills

Languages C/C++, Java, Python, C#, JavaScript

Tools PyTorch, numpy, Firebase, MongoDB, MySQL, SFML, Node.JS, React.JS, React Native