Alan Yuan

📞 +1-647-918-8482   in /JaLnYn

✉ alan.yuan.jly@gmail.com   ⏷ /JaLnYn

◔ Experience

Software Engineer - PEY Intern | Intel   C++, Python, Bash
Summer 2021 - present

› Worked in the synthesis team, working on multiple tools and new features.

› Python and Bash scripts to automate mass code changes and testing.

› Worked with program that generates Verilog which is used to test correctness of compilers.

Software Developer | Centivizer   Node.JS: SimplePeer, Socket.io, Axios
Summer 2020 - Fall 2021

› Designed and built server that connects "friends" to a video call with NodeJS and SimplePeer

› Used Socket.io to communicate between the client and backend.

› Connected the server with the database through a RESTful API using the Axios library.

🎓 Education

University of Toronto   Sep, 2018 – Jan, 2023
Candidate for Honours B.S. in Computer Science   *cGPA: 3.83/4.0*

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

⚙ Skills

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

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

⚗ Side Projects

CFR Minimization (Kuhn poker, Tic-tac-toe and Coup)   Python, PyTorch
May, 2021-Present   *github.com/JaLnYn/pokerbot*

› Worked with **CFR, CFR+ and MCCFR** to train agents to find the nash equilibrium.

› Utilized availible resources on the internet to learn these algorithms.

› Implemented each of the algorithms to play tic-tac-toe.

› Plan to use the algorithms to solve Kuhn poker and Coup.

Q-learning M.L algorithm (to play Snake)   Python, PyTorch
Dec, 2020-Jan, 2021   *github.com/JaLnYn/mlsnake*

› Read articles and papers to understand **Q-Learning**

› Implemented a Q-learning snake on top of a existing implementation of the snake-game.

CSC311 (Intro to M.L) Course Competition   Python, numpy
Nov, 2020-Dec, 2021   -

› 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.

Merchant Sensei Scraper
Python: boto3, bs4
Dec, 2019-present   *merchantsensei.com*

> Created **script to scrape the web** for HTMLs and other useful information to be run on EC2s.
> Using Python's threading capabilities, gave script ability to scale with CPU power
> Automated the packaging of the extracted data
> Sends ZIP files to AWS bucket in a nice ZIP file to minimize storage costs

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)

Web-Chat | (Hack the North)
Firebase Realtime DB, Azure M.L. API, ReactJS, JavaScript
Sep, 2019-Sep, 2019   *github.com/JaLnYn/DASH*

> Led a team of 4 to create a web-chat-app that matches strangers based on their current emotion determined by **Azure ML API**
> Built the frontend of the web-app utilizing **Firebase** and **ReactJS**
> Utilized **Firebase Realtime Database** to easily facilitate the chat rooms

Evolutionary M.L algorithm (to play Snake)
C++
Mar, 2018-Apr, 2018   *github.com/JaLnYn/Machine-learning-Snake*

> Read articles and papers to understand and implement the N.E.A.T Evolutionary algorithm with raw C++.
> Tested multiple fitness metrics such as score, survival time, and time between scoring.

Esoteric Language interpreter
C
Nov, 2017 - June, 2018   *github.com/JaLnYn/Bf-interpreter*

> Utilized recursion and memory management to implement a simple interpreter for the Esoteric Language: BF