

MITCHELL SNAITH

Site: mitchellsnaith.com Email: msnaith@edu.uwaterloo.ca Phone: 1 (403) 992-6351

SUMMARY

Quantitative software developer specializing in efficient modern C++, scientific computing, and machine learning.

EXPERIENCE

- **Klashwerks** Ottawa, ON
Software Engineer *Sep 2017 - Dec 2017*
 - **Real-time hand detector:** Built a real-time hand and gesture detector for a mobile-like platform based on the You Only Look Once paper using PyTorch and OpenCV
 - **Android media engine enhancements:** Added performance-critical functionality into Android's C++ media playback engine to allow for a greater set of real-time media data listeners
- **Citadel LLC** Chicago, IL
Quantitative Trading Systems Developer *Jan 2017 - May 2017*
 - **MongoDB latency tracking:** Designed a Python application to manage latency information on mongoDBs used by researchers, offering visualizations and forecasts of access latencies with Flask and identifying troublesome database queries
 - **Research data quality testing:** Implemented a Python and R application to allow researchers to construct and schedule quality tests on incoming data streams
- **NN Group** Amsterdam, The Netherlands
Risk Management - Quant Developer *Jan 2016 - Sep 2016*
 - **Value-at-Risk calculation engine:** Worked on various risk components of VaR engine in C++
 - **Interest rate modelling tools:** Completed prototyping tools with Python scientific libraries for interest rate models
- **RPM Technologies** Toronto, ON
Software Engineer *Jan 2015 - May 2015*

EDUCATION

- **University of Waterloo** Waterloo, ON
Bachelor of Mathematics in Mathematical Finance, Honours, Co-operative Program *Sep. 2013 - Dec. 2018*

PROJECTS

- **pytoyolo:** Efficient implementation of the You Only Look Once object detection framework paper in PyTorch.
- **exploring_kuzushiji:** Exploration, dimensionality reduction, and method survey on the Kuzushiji-MNIST dataset released in December 2018. The best method found by researchers in the paper was improved upon using Random Erasing data augmentation with a preactivated residual network in PyTorch.
- **ada_stump:** Implementation of AdaBoost with stumps using the C++ xtensor library.
- **wide_fashion:** PyTorch implementation of a 40-layer Wide Residual Network with Shake Shake regularization on the Fashion-MNIST dataset, achieving a test accuracy of 97.4 %.
- **mchess:** C++ chess engine using magic bitboards, alpha-beta pruning, and several opening heuristics. Aimed to be a building block for further engines with more creative search functions.
- **statr:** Header-only C++ library offering statistical functions. Uses template metaprogramming with Boost.Hana to find compile time dependencies between desired statistics for improved runtime efficiency.
- **complexity_sort:** Sort functions by their asymptotic complexity in Python.

SKILLS

- **Languages:** C++, Python, Haskell, Rust, SQL
- **Favorite libraries:** PyTorch, SciPy, Flask, Eigen, Boost, Scikit, Pandas
- **Technologies:** Void Linux, Emacs, TeX, Git, AWS