Eric Le Fort

Curriculum Vitae

"I strive to push myself past old limits and have found that by putting in the work I can achieve most anything."

Professional Summary

I am looking to further my career in AI.

A dedicated, hard-working, and competent engineer and researcher. I've been programming for 14+ years (7 professionally) in a wide variety of languages and have an in-depth knowledge of various ML and DL techniques. I have worked professionally with both NLP and Computer Vision.

Education

2017–2018 **Master's of Applied Science in Software Engineering** *McMaster University*, GPA, 3.8.

2013–2017 **Bachelor of Software Engineering** *McMaster University*, GPA, 3.8.

Awarded first place for my team's engineering capstone project.

Masters Thesis

Title *A Comparative Study of Machine Learning Algorithms*

Supervisors Professors Antoine Deza and Franya Franek

Abstract The selection of machine learning algorithm used to solve a problem is an important choice. This paper outlines research measuring three performance metrics for eight different algorithms on a real-world dataset. The thesis revolved around a project performed on behalf of the university to predict the likelihood of applicants accepting an offer of admission. The algorithms that were tested are k-nearest neighbours, decision trees, random forests, gradient tree boosting, logistic regression, naive bayes, support vector machines, and artificial neural networks. These algorithms were compared in terms of accuracy, training time, and inference time.

Experience

Nov 2022 – Current **Machine Learning Engineer** *Google*, Seattle, WA.

I was an SDE on the 3P Integrations and Frameworks team at Google Cloud TPU. The team was responsible for maintaining State-Of-The-Art (SOTA) model performance tests on Cloud TPUs.

Roles and Accomplishments:

- Created and maintained several TPU-based TensorFlow SOTA model nightly tests
- Designed a CLI toolkit for TPU users to better understand their training runs and debug issues
- Designed a kernel-level adaptation to support the PJRT runtime on SparseCore (proprietary embedding hardware accelerators)
- Owned the release of the official TPU Docker images for TensorFlow (versions 2.13, 2.15, 2.16)
- Helped write official technical documentation (e.g. *the TPU runtimes page*)
- Consulted with customers to help them onboard onto TPUs smoothly
- Languages: Python, C++

May 2019 – Nov 2022 **Software Development Engineer** *Amazon*, Seattle, WA.

I was an SDE on a Computer Vision team at Amazon Go. The team was responsible for a Deep Vision Inference platform, embedded Deep Vision, and two core algorithms responsible for detecting customer-product interactions. I led several major projects from design to release.

Roles and Accomplishments:

- Led a team of 3 in the design/implementation of an org-wide library that grades the performance of deep learning algorithms in an offline setting
- Led the design/implementation of a new algorithmic component that reduced prediction errors by ~1% in live systems along with a research scientist
- Led the design/implementation of a tool that automates prediction error investigation – cut down investigation time from 2 hours/error to 10 minutes/error
- Trained as a security certifier and certified an embedded computer vision component
- Created a GPU health-checking library to automatically release unhealthy EC2 hosts
- Acted as the leading Python expert on my team providing guidance on best practices
- Mentored new junior engineers on the team
- Designed and maintained complex systems running in AWS
- Languages: Python, Java, C++

Dec. 2018 – May 2019 **Machine Learning Researcher** *SortSpoke*, Toronto, Ontario.

Participated as the lead researcher in an early-stage startup. I contributed various core features to the product and I represented the company externally.

Roles and Accomplishments:

- Built a table-level information extraction system which autonomously trains models per-customer-use-case, extracts a table of interest in a document after under 20 samples
- Built a sentence-/paragraph-level information extraction system which is also autonomously trained per-customer-use-case
- Built a checkbox OCR feature using OpenCV
- Developed the ML R&D strategy and consulted with the business team regarding ML capabilities
- Assisted in the recruiting and hiring process
- Languages: Python (PyTorch, Scikit-Learn, NumPy), C#/.NET, MongoDB

May. 2018 – Dec. 2018 **Data Science Engineer** *AbacusNext*, Toronto, Ontario.

Helped spearhead a new team focused on data science initiatives.

Roles and Accomplishments:

- Designed and implemented a document anonymization system powered by NER
- Designed and implemented a document topic modelling system
- Built Azure cloud infrastructure to host production systems
- Optimized a critical system resulting in a 90% speedup
- Supervised and mentored engineering interns
- Participated as an interviewer in the hiring process for data scientists, machine learning engineers, and data science QA analysts
- Languages: C#, Python, NoSQL databases

Oct. 2017 – Feb. 2019 **Machine Learning Consultant** *Self-Employed*, Various Locations .

With an academic research team, I converted a deep learning CNN image algorithm called LSDA (http://lsda.berkeleyvision.org/) from MATLAB/Caffe to Python/PyTorch.

With Old Republic Insurance, I analyzed sales data (which I augmented by scraping online census reports) using unsupervised learning techniques.

With Hazen & Sawyer, I performed both hands-on work and technical mentoring. The project was a time-series analysis of key metrics in waste water systems.

Summer 2017 **Machine Learning Engineer Intern** *Sensibill*, Toronto, Ontario.

This position involved participating as a researcher in a small R&D team at Sensibill.

Roles and Accomplishments:

- Improved a mission-critical OCR (Optical Character Recognition) process
- Developed an LSTM-based language classifier as part of an internal research effort that achieved 98.9% accuracy.
- Scaled a key data pipeline to handle arbitrarily large datasets efficiently

Sept. 2016 – Apr. 2018 **Graduate Teaching Assistant** *McMaster University*, Hamilton, Ontario.

I assisted teaching courses covering topics such as: optimization, queuing theory, statistics, bash scripting, Markov chains, Petri Nets, etc. I ran weekly labs, merked, and taught tutorials.

May 2016 – Sept. 2017 **Research Assistant** *McMaster University*, Hamilton, Ontario.

Supported the research of Dr. Jeffery Zucker and Dr. Antoine Deza at different times.

Programming Languages

Advanced	Python, Lua, Java, C#, Bash
Intermediate	Scala, R, C++, JavaScript, SQL, NoSQL, MATLAB, Rust, HTML5/CSS
Basic	C, PHP, Haskell, Swift, TypeScript, OCaml, Prolog

Programming Skills

Competencies	Convex Optimization Algorithms; Calculus; Database Design; Propositional Logic; Distributed Computing
Machine Learning	NLP; Neural Networks such as MLPs, RNNs, CNNs; PCA; Transfer Learning; Synthetic Data Generation; SVMs; Random Forests; K-NN; Naive Bayes; and more
Tools	Torch, PyTorch, TensorFlow, Keras, Scikit-Learn, Numpy, Pandas, Git/SVN

Projects

LSTM Language Model This project involved testing the difference between a word- or token-based language model against a character-based model. I applied rigorous scientific techniques to arrive at this final conclusion.

Notable Components:

- Utilized Torch (Lua's ML-library) to quickly develop prototypes.
- Fine-tuned parameters and the learning algorithm using various techniques to maximize model performance.
- Achieved a final accuracy of 98.9% (the majority of this error arose from very similar languages such as Spanish and Italian).

Capstone For our capstone, my team made an autonomous billiards robot. A rigorous requirement gathering, design, and testing process was followed to ensure successful implementation. Development languages include Java, MATLAB, C++, and C. This machine was built with a 6 person interdisciplinary team. This project was also selected as the department's best in show for the 2017 year.

Notable Components:

- Utilized vision recognition to locate and identify pool balls.
- Defined a custom algorithm to select an optimal shot.
- Created a pool physics simulator from scratch.
- Communicated between an Arduino, a PC, and an Android device.
- Implemented a software controller for the robot.

GitHub Repository: github.com/eric73837/Capstone-Project

PokerBot The end goal for this project is a poker-playing A.I. that wins at least 51% of the time against professional poker players. The first phase of this project is creating a framework to be used by the rest of the program. This allows for the program to interface with the "world" separately from the more complicated components. The second phase will be creating a rule base to provide fundamental principles of how to play the game of poker well. The third phase will be using the framework and rule base to implement the A.I. itself.

After each phase, a thorough testing suite will be created to verify the code. For the third phase, this testing will mainly consist of performance monitoring.

Notable Components (chronologically):

- Effective Poker data structures.
- Parsing ASCII text files containing hand histories.
- Constructed a basic Poker logic inference engine to define the rules of the game.
- Statistical situation analysis (in progress).
- Machine Learning–maximize profit according to Poker Game Theory (in progress).

GitHub Repository: github.com/eric73837/PokerBot