Eric Le Fort

Curriculum Vitae

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

Professional Summary

Pursuing a software engineering career in the field of Machine Learning.
An extremely dedicated, hard-working, and competent software engineer. Notable abilities include 7+ years of experience programming in various languages and in-depth knowledge of various machine learning techniques. Ideal positions would involve a focus on applied machine learning or data science.

Education

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

My specialization is in the field of Machine Learning. My thesis revolves around a project performed on behalf of the university to predict the likelihood of applicants accepting an offer of admission. I plan to graduate in May 2018.

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

Awarded best in show in the 2017 graduating class for my engineering capstone project.

Experience

Oct. 2017 – Apr. 2018 **Machine Learning Consultant** *Old Republic Insurance*, Hamilton, ON.

This contract involves analyzing geographically segmented customer data to uncover better market clusters and other valuable business insights. Publicly available datasets were also used to bolster the information available. The analysis was performed using various unsupervised learning techniques.

| Summer 2017 | **Machine Learning Engineer Intern** | *Sensibill*, Toronto, ON. |

This position involved participating as a member of a small research and development team within Sensibill.

Accomplishments:
- Worked on an OCR (Optical Character Recognition) process
- Developed an LSTM-based language classifier
- Scaled a key data pipeline to handle arbitrarily large datasets efficiently

| Sept. 2016 – Current | **Graduate Teaching Assistant** | *McMaster University*, Hamilton, ON. |

I have assisted in teaching various courses: "Digital Systems and Interfacing," "Software Development Skills," "Performance Analysis of Computer Systems," and "Operations Research". These courses covered topics such as linear optimization, Markov chains, Petri Nets, statistics, queueing theory, Linux scripting, assembly code, hardware/software co-design, application specific processors, and interfacing with I/O devices.

Roles:
- Running and invigilating weekly labs.
- Marking of labs, tests, and the final exam.
- Marking major labs and final projects.
- Communicating with students.

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

I had two supervisors for this position: Dr. Jeffery Zucker and Dr. Antoine Deza. My first project was to prepare course material for ?2FA3: Discrete Math with Applications II? using LaTeX. For the second project, I architected an analytical system utilizing machine learning to predict which applicants would accept offers of admission to the university.

Programming Languages

Advanced	Java, Lua, LaTeX, Bash
Intermediate	Python, Scala, R, C++, JavaScript, MySQL, 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	Neural Networks (LSTM, Convolutional), Kernel PCA, Support Vector Machines, Random Forests, K-Nearest Neighbour, Naive Bayes Classifiers, and more
Tools	Torch, TensorFlow, Git/SVN, Maple, VIM, Emacs, Android Studio

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 a 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
In summary, this project implemented 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

Interests

- Machine Learning
- Game Theory
- Optimization of Resource Allocation
- Mathematical Foundations of Optimization Algorithms
- Code for Change
- Automated Code Optimization
- Behavioural Psychology