Gary Mindlin Miguel

garymm@garymm.org, https://www.linkedin.com/in/gary-miguel/

Summary

Software engineer and people manager. I want to push AI forward. Experienced on everything from huge teams on very mature projects (e.g. Google search) to tiny teams doing extremely exploratory research (e.g. Obelisk). I try to improve the productivity of every team I'm on through technical, process and cultural means.

Work experience

July 2022 - present: Obelisk AI research at Astera Institute

The Astera Institute is a research institute focused on high-leverage ways to increase scientific and technological progress. Obelisk is an AI research lab at Astera mostly focused on problems in a reinforcement learning context (i.e. agents interacting with an environment).

- 2023 - present: Vice President of engineering for Obelisk. Responsible for hiring and managing software engineers and research engineers. Responsible for all layers of infrastructure, from personally assembling a small GPU cluster to designing interfaces to make our researcher's workflows easy and fast to execute on our clusters. Built a framework to reduce boilerplate and enable high performance reinforcement learning training using JAX. Worked with CEO and scientists to set research agenda.
- 2022 - 2023: Software engineer on Obelisk. Implemented reinforcement learning environments and integrated them with our model training code. Ran experiments comparing open-source state-of-the-art vs our stuff on image classification. Implemented a distributed hyperparameter search system. Learned a bit about neuroscience.

March 2021 - July 2022: AI Frameworks at Microsoft

The AI Frameworks team at Microsoft built software that is aimed at making machine learning models faster (both during training and inference) and more portable. The team was the largest contributor to the Open Neural Network Exchange (ONNX) ecosystem and the owner of ONNX Runtime. ONNX is a specification of machine learning models that decouples them from any particular framework (e.g., PyTorch) or hardware platform (e.g. Nvidia CUDA).

- Principal Software Engineer. Tech lead and manager of a team (10 engineers) working on the ONNX ecosystem. My scope included converters (sort of like compilers) from PyTorch and TensorFlow into the ONNX format, the format specification, and tooling. Implemented major changes to dev process to increase happiness and productivity. Worked with hardware vendors to design features to further accelerate machine learning training and inference. Improved usability through docs and code. Some of the changes I reviewed, authored.

January 2018 - March 2021: Fuchsia Operating System at Google

Fuchsia was an open source operating system, including a new kernel, file systems, network stack, etc. As of 2024, it has been deployed on Google's Nest Hub smart displays.

- 2018-2021: Staff Software Engineer. Tech lead of platform infrastructure team. Developed systems to automate building and testing of the OS on developer workstations and in a distributed CI system on VMs and bare metal test devices. Scaled from ~100 developers and 1 supported device type to ~500 developers and ~5 device types. Some of the changes I authored or reviewed.
- 2019-2020: 20% volunteer on Debug at Verily. Migrated mosquito image classification pipeline off of Google Cloud proprietary tech to RabbitMQ so it could run on-prem.

June 2010 - Dec 2017: Search Infrastructure at Google

The search infrastructure teams at Google built the systems that power search across many Google products, including Google Search, YouTube, and Maps. This included offline data processing (crawling

the web, processing documents to extract features, building an index, copying petabytes of data around) and online serving (i.e. actually executing the search).

- 2015-2017: Senior Software Engineer. Tech lead and manager of the engineering productivity teams (9 engineers in 2 locations) for the crawl and indexing teams. Provided infrastructure for automated black-box and end-to-end testing, release automation, experimentation and developer productivity.
- 2010-2015: Software engineer. Individual contributor working on engineering productivity. Developed systems for testing and experimentation with Google's largest distributed systems (search indexing and serving), scaling systems down from a data center to a developer workstation while maintaining fidelity.

July 2009 - June 2010: Informatica

Software Engineer. Developed Eclipse-based tool used for developing data services based on integrated data from multiple systems.

Formal education

2011-2016: MS in Computer Science, Stanford University

Pursued concurrently while working at Google. Specialization in systems, including operating systems, distributed systems, networking, databases, and computer architecture: https://cs.stanford.edu/academics/masters/choosing-specialization#systems

Electives focused on data analysis, including machine learning.

2005-2009: BA in Computer Science, University of California, Berkeley

I received the 2009 award for Outstanding Academic Achievement, which is given annually by the Computer Science department to the student with the highest GPA in the graduating class.

Skills

Machine learning: Know the basics of the theory of common methods (e.g. MLP, transformer, CNN, RNN, DQN). Actually trained models for experiments in reinforcement learning and supervised learning (image classification). Built reinforcement learning environments.

Software engineering domains: Developer infrastructure (test automation, continuous integration, release automation, build systems like CMake and Bazel), data analysis (SQL, batch processing / MapReduce), distributed systems (Google Cloud Platform, Kubernentes, RabbitMQ, many Google-internal technologies), machine learning frameworks (PyTorch, JAX, ONNX, Ray, Gymnasium), provisioning and configuring servers, network configuration (firewalls, network booting, etc).

People management: 5 years experience as a manager of software engineers and research engineers. Hired (from writing job descriptions to interviewing to negotiating offers), coached, refined processes, evolved team culture, evaluated performance, set compensation, managed intra- and inter-team conflicts.

Computer languages: C, C++, Go, Python, SQL.

Human languages: English (native), Spanish (basic), Mandarin Chinese (very basic).

Self-teaching

Some books which have influenced the way I work: Code Simplicity, A Philosophy of Software Design, A World Without Email, Nonviolent Communication.

Some resources I have found useful for learning about Machine Learning: Practical Deep Learning for Coders, A Code-First Introduction to Natural Language Processing, Yannic Kilcher, The Matrix Calculus You Need For Deep Learning (my corrections).