

Jyotinder Singh

hello@jyotinder.com • New Delhi, India • [LinkedIn](#) • [Github](#)

WORK EXPERIENCE

Keras	Google	Feb 2025 - Present Bangalore, India
-------	--------	--

Software Engineer III

- Designed and built an end-to-end benchmark framework for Qwix, which is an open-source model quantization library for JAX.
- Leading the Keras model optimization charter.
- Designed and implemented the core Keras Quantization API, an extensible framework allowing users to optimize memory footprint and compute requirements for large models
- Implemented advanced low-bit quantization techniques (INT4, GPTQ, AWQ) in Keras and Qwix, reducing model size by 80% and VRAM usage by 50% while increasing token throughput by 48% with negligible accuracy loss.
- Architected an end-to-end automated pipeline for KerasHub, eliminating all manual overhead to facilitate 100K+ annual downloads for Google internal teams and saving hundreds of engineering hours.
- Designed and implemented migration paths from TensorFlow to JAX using Keras to improve adoption of next-gen machine learning frameworks for several teams.

Ads Data Layer

Jul 2022 - Jan 2025

Bangalore, India

Software Engineer III

- Built the compiler, schema engine, and planner components for a Relational Online Analytical Processing engine using C++.
- Rearchitected the cube schema engine to support Schema Modularisation, allowing partitioning of the schemas into discrete, independently managed modules with their own namespaces, facilitating easier maintainability.
- Led work on a schema presubmit system, enabling controlled launch of schema entities to production environments - ensuring safe rollouts and deployment flexibility.
- Hosted interns and helped them succeed in their projects, promoting their growth as engineers.

Software Engineer II

- Built several core features for the schema engine such as Generated Fields, User Defined Functions, and SQL Module support.
- Built and maintained Java libraries and tools to allow users to access and interact with the cube schema to improve developer experience and productivity.

Storage Platform

Cloudera

Jan 2021 - Jun 2022

Bangalore, India

Software Engineer

- Worked on performance engineering for Apache Ozone - a distributed object store for Hadoop written in Java.
- Engineered File System Optimization techniques to improve Ozone's performance as a file system.
- Designed and implemented upgrade paths for File System Optimized Buckets, ensuring safe system upgrades.
- Developed a Cloud Cost Management Platform from scratch, enabling real-time monitoring and management of cloud resource consumption. The tool helped reduce the organisation's average cloud spend by more than 20%.

EDUCATION

-
- | | |
|---|-------------|
| ● B.Tech. Information Technology - Manipal Institute of Technology, India. | 2017 – 2021 |
| - GPA: 8.86 / 10 (Three times gold medalist) | |
| ● High School Diploma - Bal Bharati Public School, New Delhi, India. | Grad. 2017 |
| - Class 12 th Percentage (CBSE): 92.4% | |

TECHNOLOGIES AND LANGUAGES

- Languages: Go, C++, Java, Python, TypeScript, JavaScript
- Technologies: Keras, JAX, TensorFlow, Docker, gRPC
- Other: Compilers, Database Systems, Algorithms, Data Structures

PROJECTS

- **DiceDB - A reactive, scalable, and unified cache optimized for modern hardware.** [\[Link\]](#)
 - Co-creator for DiceDB, directing the core vision of the project and managing a massive open-source community.
 - Built and architected core data structures, reactive subscriptions, and the multi-threaded engine.
 - Designed the observability stack, client SDKs, and a comprehensive test suite.
- **dropdb - A From-Scratch Relational Database in Go** [\[Link\]](#)
 - Developed a fully functional relational database engine.
 - Implemented disk/file management, memory management, transaction handling, and query planning.
 - Added support for Go's sql interface standard, for familiar user experience.
- **goLSM - Write-Optimized Embedded Database** [\[Link\]](#)
 - Built a Log-Structured Merge Tree based write-optimized embedded database from scratch using Go.
 - Benchmarked using Yahoo Cloud Storage Benchmark and provides ~10 μ s reads, ~10 μ s writes, ~20 μ s updates.
- **Apache Ozone - Committer** [\[Link\]](#)
 - Committer in the Apache Ozone open-source project, which is a scalable, redundant, and distributed object store for Hadoop.
- **Tok - Programming Language** [\[Link\]](#)
 - Wrote a Virtual Machine in C for a programming language called Tok.
 - Supports dynamic typing, automated garbage collection, flow control statements, user-defined functions, closures, classes, and inheritance.
 - Wrote a Remote Code Execution engine [\[Link\]](#) which hosts the compiler on the web and allows users to try out the programming language without having to install anything on their system.

PUBLICATIONS

- **YouTube Channel** [\[Link\]](#)
 - I run an educational YouTube channel with more than 5000 subscribers, creating comprehensive videos related on building complex software systems from scratch, system design, and algorithms.
- **Newsletter** [\[Link\]](#)
 - I publish a newsletter focused on Machine Learning, databases, and distributed systems read by hundreds of subscribers every month.