# Junshang Jia

zohargates@gmail.com | 412-403-0254 | linkedin.com/in/junshang-jia | github.com/zoharrpg | Pittsburgh, PA

## Education

M.S. in Computer Science(System), **Carnegie Mellon University** Aug 2023 – Expected May 2025
B.S. in Computer Science, **University of Pittsburgh** GPA: 3.9/4.0 Aug 2019 – May 2023

## Skills

**Programming Languages**: Java, C/C++/STL, CUDA, Python(PyTorch,Pandas), Go, TypeScript, JavaScript, SQL,HTML/CSS
**Frameworks**: Spring Boot, NestJS, Fastify, GraphQL, React.js, Node.js, Angular, NoSQL, MongoDB, PostgreSQL, MySQL
**Tools**: Linux, Git, Bash/Shell, AWS, Azure, GCP, Docker, Kubernetes(K8s), Github Actions, Terraform, Jenkins
**Knowledge** OOP/OOD, Design patterns (MVC, Factory, pub/sub, etc.), Agile, Microservices, RESTful API, CI/CD

## Work Experience

**Capital One** Software Engineer Intern Jun 2024 – Aug 2024
NestJS based Web Service for Identity Management Dallas, TX
- Developed a company-wide system using NestJS, Fastify, and TypeScript, and deployed it to AWS Elastic Beanstalk.
- Built authentication using Fastify JWT and OAuth2, and designed authorization guards to enforce role-based access control.
- Created GraphQL APIs using Apollo GraphQL, and defined data models using Prisma ORM with PostgreSQL.
- Architected a frontend with Angular, integrating D3.js and Chart.js to create task progress visualizations.
- Constructed Jenkins CI/CD pipelines with Docker, and orchestrated deployment with Kubernetes and Terraform on AWS.

**Ericsson** Software Engineer Intern Jun 2021 – Sep 2021
Spring based Web Service for Product Lifecycle Management Beijing, China
- Spearheaded the development of the system, employing Microservices architecture and API Gateway pattern.
- Architected backend with Spring Boot, integrated authentication with Spring Security, and used Redis for caching.
- Implemented the notification services using Kafka, and utilized Streams API to process and transform notification messages.
- Innovated frontend interfaces using React, TypeScript, and Antd UI library to create product lifecycle visualizations.
- Built the Azure CI/CD pipeline with Docker and deployed the application image on Azure App Service.

## Projects

**Deep Learning Framework with Customized GPU and CPU Backend** Jan 2024 – Jun 2024
- Developed a deep learning framework and implemented automatic differentiation and tensor acceleration backend.
- Engineered and refined computing libraries for both CPU and Nvidia GPU backends using C++ and CUDA.
- Programmed classic deep learning models, including MLP-ResNet, ResNet9, and RNN/LSTM, using pybind11 and OpenMP.
- Accelerated training time by 50% by integrating OneDNN and CuDNN, providing optimized convolution operations.

**ELT and Machine Learning Training using Apache Spark** Jan 2024 – Apr 2024
- Initiated a distributed program using Apache Spark and PySpark to preprocess a large web crawl dataset.
- Monitored memory usage, task execution time, and data shuffling through the Spark web interface.
- Created a scalable distributed machine learning program using Spark RDD, reducing training time by 60%.
- Deployed a Spark cluster on AWS EC2 and automated its provisioning using Flintrock to achieve an efficient setup.

**Golang Based Serverless API** Sep 2023 – Dec 2023
- Architected RESTful API endpoints with Go and AWS Lambda, providing backend services for forum data management.
- Integrated AWS services DynamoDB, S3, and API Gateway to create serverless CRUD APIs for forum data using Go.
- Monitored API performance using AWS CloudWatch and X-Ray, enabling quick identification and resolution of issues.
- Constructed Google authentication using Go OAuth 2.0 and integrated AWS Cognito for user management.

**Golang LSP Based Bitcoin Miner** May 2024 – Jul 2024
- Engineered a scalable system using a custom Live Sequence Protocol (LSP) to perform bitcoin mining with Go.
- Introduced a time-triggered epoch system, bolstering robustness through transmission of heartbeat messages via channel.
- Optimized a load balancing system to ensure efficiency and equitable task allocation based on request size and arrival time.
- Applied the acknowledgment protocol using gRPC for streamlined communication.