Github.com/Michael-Nath

Michael Nath

mnath@stanford.edu

Education

Stanford University, Stanford, CA
B.S. Mathematics and Computer Science, GPA: 4.0 09/2021 – 06/2025
* *- graduate level courses*

Courses: Matrix Theory, Blockchain Foundations*, Neuroelectrical Engineering, Operating Systems, Sequential Decision Making*, Computational Biology*, Reinforcement Learning*, Natural Learning Processing with Deep Learning*

Stuyvesant High School, New York, NY 09/2017 – 06/2021

Professional Experience

Amazon Web Services (AWS AI), Seattle, WA 06/2022 – 09/2022
Software Development Engineer Intern

- Tasked by the Anomaly Detection team to contribute infrastructure that automates cleanup of extraneous cloud resources
- Automated patching and recycling of compute resources used by customers, saving on-call engineers 36+ hours biweekly
- Helped DevOps engineers achieve continuous development by adding scheduled, automatic triggers to team pipelines
- Invented an on-demand clean-up algorithm enabling dynamic compute allocation, redesigning 2 cloud workflows used daily
- Hacked Python to digest metrics of pre- and post-cleanup performance, which proved new workflow benefits to team
- Assisted senior engineers after project completion to patch up other resources scattered across miscellaneous workflows
- Employed: AWS Lambda, EC2, S3, CloudFormation, DynamoDB, Step Functions, EventBridge, Kotlin, Mockito

Brookhaven National Laboratory, Upton, NY 06/2021 – 08/2021
Computational Scientist Intern

- Tackled the optimal qubit control task by generating optimal pulse coefficients through deep reinforcement learning
- Practiced quantum computing engineering from following Qiskit documentation and parsing relevant literature
- Gained familiarity with professional neural network architecture, learning best practices in modern research and development
- Co-authored robust qubit gym environment leveraging epsilon-greedy policy and temporal difference learning to train agent
- Improved quantum control fidelity by 10% by trying deep q network weights as pulse coefficients as moonshot idea
- Elucidated training results to department heads through interactive plots and demos, comparing tabular and deep performance
- Employed: PyTorch, OpenAI Gym, NumPy, Deep Q Learning, Tabular Q Learning, Matplotlib

Projects

EZCar – *Simplified Car Search Platform*

- Led team of 4 engineers under mentorship at Google to streamline car search for consumers through beginner-friendly app
- Aggregated data in SQL database of 1000+ modern cars by scraping kbb.com and cars.com via Python bot
- Handled simultaneous requests of cars meeting 5+ filters and 12+ categories through RESTful Flask server
- Invented innovative gold-silver-bronze badge system ranking 100+ filtered cars state-fully with React
- Employed: SQLAlchemy ORM, BeautifulSoup4 Scraper Library, Material-UI, Axios, Python, JavaScript

Adversary – *Full Node on the Marabu Blockchain*

- Theorized and coded an UTXO-based full node from scratch on a TCP network protocol with 30+ other full nodes
- Implemented peer-to-peer discovery, proof-of-work, recursive chain validation, mempool reorganization, static difficulty
- Gained knowledge in engineering distributed networking, blockchain data structures, and in-memory data storage
- Employed: Typescript, Node.js, Net, LevelDB, stream, SHA-256, ED25519

WorldView – *NLP-Driven Perspective Comparator and Synthesizer for News Articles*

- Hacked together chrome extension presenting diverse perspectives of topics in news articles powered by sentiment analysis and named-entity recognition
- Designed Flask and Express API to create pipeline between core NLP processor + web scraper and React frontend
- Awarded *Best Collaborative Hack* by VMWare at Stanford TreeHacks for orchestrating machine learning, cloud computing, and extension development
- Employed: Express, Flask, Python, GCP Search API, KProtoype Clustering, Chrome Extensions, NetworkX, NLTK

Stuyvesant Science Bowl Discord Bot – *Virtual Science Bowl Match Assistant*

- Designed robust buzzer system featuring unlimited game modes, notepads, and buzzer analytics for 20+ players
- Set up simultaneous matchmaking through manipulating global hash maps, allowing for maximal game time
- Implemented PDF parser to scrape questions from tournaments to fuel future auto question-reading feature
- Employed: Node.js, Sequelize, Discord.js, JavaScript