

Spencer Stice

smstice17@gmail.com 720.376.3805 [linkedin.com/in/spencer-stice](https://www.linkedin.com/in/spencer-stice) <https://spencer-stice.github.io>

Education

University of California, Los Angeles (UCLA) - Henry Samueli School of Engineering Los Angeles, CA
Bachelor of Science in Computer Engineering and Mathematics minor, GPA 3.894 October 2020 - June 2024 (Expected)
Honors: IEEE-Eta Kappa Nu Honor Society Membership Chair (top 25% of class), Dean's Honor List
Relevant Coursework: Deep Learning 1 and 2, Computer Vision, NLP, AI Theory, Algorithms, Operating Systems

Skills

Programming Languages: Python (Experienced), C++/C (Experienced), SystemVerilog (Experienced), Java (Intermediate), JavaScript (Intermediate), Dart (Beginner)
Technologies/Concepts: Tensorflow, PyTorch, Transformers, LSTM/RNNs, CNNs, GANs, Genetic Algorithms, SIFT, Kalman Filters, Matplotlib, Numpy, Pandas, Git, Linux, Google Firebase, React, VS Code, Postman, Intel Quartus Prime, ModelSim, BERT, Oscilloscope

Work Experience

AI Research Intern: Perceptronics Solutions El Segundo, CA
Skills: Tensorflow, Neural Networks, Control Theory (Kalman Filters), Research June 2023 - September 2023

- Explored the practical applications and effectiveness of AI/ML solutions, including transformers, LSTMs, and particle filters in addressing critical challenges associated with positioning cards, while conducting a review of research papers
- Designed and implemented a NARX neural network utilizing Tensorflow to address a prominent issue in the system
- Performed simulations on the modified positioning system, demonstrating a 55% improvement over the original system

Software/Memory Engineer Intern: Intel Corporation Folsom, CA
Skills: DDR5, Python, Genetic Algorithms, ReactJS, SQL June 2022 - December 2022

- Developed a Python-based AI program utilizing genetic algorithms to optimize duty-cycle adjust settings, resulting in an 80% reduction in DDR5 memory validation times and the discovery of superior configurations
- Conducted over 13,000 setting distribution tests to assess the program's performance potential, leading to the submission of an invention disclosure form
- Initiated the overhaul of the Memory/IO team's outdated database website using ReactJS, FastAPI in Python, and SQL
- Developed a Python parser program for LPDDR5 memory, enabling the team to efficiently extract crucial information from the tested memory modules

Projects

Quill: ChatGPT-powered Chrome Extension
Skills/Technologies: OpenAI API, HTML/CSS/JS, Chrome Extension manifest.json files

- Led a team of 4 to create a Google Chrome extension that integrates ChatGPT into the web browser
- As CEO of our startup, pitched to co-founder of Google AdSense and other prominent LA VCs
- Attained over 700 users during early development; voted best pitch/idea by AdSense co-founder

Microsoft Phi-2 Language Model Fairness Experiments
Skills/Technologies: Language models, transformers, Google Cloud Compute

- Designed a test suite to evaluate the fairness of the Phi-2 transformer-based language model
- Used the Huggingface transformer library and Google Cloud compute to run inference and evaluate results

EEG 4-Class Motor Classification via Recurrent and Convolutional Networks
Skills/Technologies: PyTorch, Recurrent NNs, CNNs, Data Preprocessing

- Designed and implemented various NN architectures to classify EEG data into 4 classes of motor activity
- Performed hyperparameter tuning and experimentation with various architectures to optimize performance
- Achieved 69.3 % test accuracy using a convolutional network with L2 regularization

Digital Audio Visualizer (DAV): FFT Display
Skills/Technologies: SystemVerilog, Signal Processing

- Developed and implemented a 32-sample Fast Fourier Transform (FFT) algorithm on an Intel Max 10 FPGA, enabling real-time audio signal processing
- Utilized a digital microphone for audio data acquisition and a VGA monitor for displaying the FFT results, creating an interactive audio visualization tool
- Won first place in the project competition among 15+ teams
