Daniel Zeng

danielzeng@berkeley.edu (408) 987-1872 github.com/daniel-zeng
linkedin.com/in/daniel-zeng danielzeng.com

University of California, Berkeley *May 2021 (Expected)*
Computer Science, Bachelor's Degree GPA 3.97

Relevant Courses (*enrolled)
CS189* - Machine Learning, CS170* - Algorithms, CS188* - Artificial Intelligence,
CS61B - Data Structures, CS70 - Discrete Math and Probability Theory, CS61A - Interpretation of Computer Programs, MATH54 - Linear Algebra

Skills
Languages: Python, Java, C++, Javascript, GNU Octave
Platforms/Tools: numpy, tensorflow, pytorch, matplotlib, Git, Bash, pandas, requests, sockets, JUnit

Experience

Software Engineering/Research Intern — NASA Ames Research Center *June 2018 - Aug 2018*
- Developing an emulator for cyber security attacks on Air Traffic Management (ATM) system
- Designed and built a modular and extensible architecture on the ATM Testbed framework
- Implemented and tested functionality to generate wide-range of attack scenarios

Project Developer — Cal Launchpad, UC Berkeley *Sept 2017 - Present*
- Project Facelift: Implemented stacked convolutional neural networks in Tensorflow to compute facial depth from 2D image
- Utilized dlib to preprocess facial features and perform facial alignment

Research Assistant — Dal Bó Lab, UC Berkeley *Sept 2017 - May 2018*
- Developed algorithms for natural language processing and statistical analysis to identify past civilization trends
- Built methods for collecting archaeological articles using Crossref API and web scraping
- Implemented methods for text corpus processing to extract relevant information

Research Intern — Bhatia Lab, Boston University *July 2016 - Aug 2016*
- Accelerated and automated the process of designing genetic circuits (synthetic biology) using machine learning
- Created web interface using Vaadin to allow users to generate desired circuit functions

Projects

DeepBeat (Cal Launchpad) *Sept 2017 - Dec 2017*
- Researched a novel neural network architecture — phase functioned LSTM model
- Built methods for data processing on music MIDI files to utilize during network training

Chess Game (github.com/daniel-zeng/ChessGame) (Python, tkinter)
- Implemented minimax tree search with Alpha-Beta pruning, and position evaluation functions (piece square tables/piece values) to optimize the AI engine
- Playable through command-line with algebraic notation and GUI (tkinter)