Tony Pan

(908) 448 5096
tonyliupan@gmail.com
linkedin.com/in/tonylpan
tonylpan.com

education

University of California, Berkeley — *Aug 2015 — May 2019*
Computer Science B.A.
Major GPA — 3.5

experience

Adaptive Insights, A Workday Company — *June 2018 — August 2018*
Software Engineering Intern

- Assisted in implementing proof-of-concept AI/ML service in Java
- Built Anomaly Detection API, with Spring Boot RESTful API endpoint
- Added protocol buffer support to existing service API
- Created time-series processing and aggregation tools in Spark and Pentaho Kettle for data science ETL pipeline
- Connected Cassandra databases to company services through ORM layer implemented using Spring Data Cassandra

skills

Python, Java, C, C++, SQL, Git, Spark, Gradle, Spring Boot, Spring Data

projects

MNIST Digit Classifier — *Python*

- A Python neural network to classify images of handwritten digits
- Optimized on 60,000 images from MNIST digit database
- Achieved overall accuracy of ~98% on test image dataset
- Built without using pre-existing machine learning libraries, using only NumPy for efficient matrix operations

Markov Tweets — *Python*

- Generates pseudo-random text by parsing data from twitter
- Utilized Tweepy API to extract text metadata from tweets on a specified user's twitter feed
- Decomposes text into word sequences to generate Markov chain
- Creates probabilistically random strings of words with Markov chain given an initial seed word

Text Editor — *Java*

- Implemented in Java, with basic notepad-esque capability
- Supports reading, editing, and creating plaintext files
- Performs text insert & delete operations in constant O(1) time, and text rendering operations in linear O(n) time
- Implements word wrap functionality that dynamically updates with window resizes, font size changes

BearMaps — *Java*

- Built a mapping application for Data Structures class, using OpenStreetMaps map tile images and coordinate data
- Utilized a quadtree data structure for efficient map rendering
- Map allows panning, zoom in/out, dynamic levels of detail
- Used coordinate data to enable plotting and route search, A* algorithm used to determine the shortest route between points
- Implemented location searching and query autocomplete using a trie data structure