Last updated in December 2024

Allan Zhang

31 E Funderburg Blvd, Mountain House, CA   allanzhang440@gmail.com   allanzhang@g.ucla.edu
609-943-8429   https://giyushino.github.io   giyushino

Education

University of California, Los Angeles *Sept 2024 – June 2028*
BS in Applied Mathematics

- GPA: 4.00
- **Relevant Coursework:** Multivariable Calculus, Linear Algebra, Discrete Structures, Computer Science (Math 32A, Math 32B, Math 33A, Math 33B, Math 61A CS 31)

Self Study *July 2024 - Present*
Machine Learning and Data Science

- **Textbooks:** *Reinforcement Learning: An Introduction*, *An Introduction to Statistical Learning with Applications in Python*, *Dive into Deep Learning*
- **Courses:** Andrew Ng's Machine Learning Specialization

Experience

Undergraduate Researcher | BigML *Los Angeles, CA*
UCLA *Nov 2024 – Present*

- Set up experiments involving vision transformers, 2D projection layer (MLP), and LLMs to study why VLMs perform poorly on relatively simple tasks
- Used mechanistic interpretability to extract tensors from every layer, analyzed them to understand how each layer processes input data. Used Seaborn and Matplotlib to create heatmaps, showing model's predictions for every input token
- Wrote functions to extract attention heads and visualize the weights each head in every layer assigns to individual input tokens

Projects

Doodle Guesser *MyOwnDoogleGuesser*

- Using vision transformer (OpenAI's CLIP-Vit-Large-Patch-14), downstreamed model to predict animal drawings created by users. Collected and processed 1,000,000+ images to create dataset to fine-tune. Original model had 54% accuracy, fine-tuned had 87%
- Wrote CLIP model from scratch using PyTorch, trained on same dataset. Achieved 70% accuracy after being trained on 200,000 training images. Also wrote functions to split dataset into batches before inference, perform data analysis to compute model accuracy compared to ground truths, and fine-tune the model. Replicated PyTorch's built in softmax and cross entropy loss functions only using NumPy
- Tested model on zero-shot classification by having users draw an animal, AI predicts what the user doodled
- Tools Used: Python, PyTorch, Pygame, Hugging Face, NumPy

Efficient Finetuning Pipeline *Fine-Tuning Functions*

- Created functions to more efficiently train weights and biases of models running on weak GPUs or CPU. Functions focused on batching images fed into model to reduce VRAM requirements and preventing Google Colab from crashing
- Tested on OpenAI's CLIP-Vit-Large-Patch-14 model using CIFAR-10 dataset. Saw 5% improvement in accuracy (91% → 96%) with limited training set, computational power, and time. To prevent overfitting, wrote new function to shuffle training dataset for every epoch
- Tools Used: Python, PyTorch, Google Colab, Hugging Face

Hand-Controlled Drone *Drone Control*

- Created and labeled dataset consisting of 400+ images of hands pointing up, down, left, right, backwards,

- and forwards. Trained YOLOv8 model with the data and utilized OpenCV to detect directions real time. Fed real-time collected information to FPV drone's flight computer to control movement
- Tools Used: Python, OpenCV, YOLOv8

Flappy Bird AI *Work in Progress*

- Created own version of Flappy Bird in Python using Pygame. Currently creating neural network that utilizes Q-learning to learn to play the game. Game state is collected at every frame is collected and fed into model, including bird's vertical position, vertical velocity, and distance from both pipes
- Tools used: Python, PyTorch, Pygame

Bathroom Usage Prediction *UCLA Bathroom* ↗

- Implemented linear regression model from scratch in Python to predict how many people are using the Rieber Hall 7N men's bathroom based on time and day. Wrote the cost, gradient, and gradient descent functions only using NumPy
- Tools used: Python, NumPy, Matplotlib

Self Published Novel *2020 - 2021*

- Wrote a 45,000 word novel, published it online. Amassed 272,000 reads, 7,000 comments, and 6,200 reviews. Reviewed other authors' works, providing grammatical advice

Skills

Programming Languages and Frameworks: Python, PyTorch, NumPy, Matplotlib, TensorFlow, scikit-learn, OpenCV, Hugging Face, LaTeX, MATLAB, HTML (basic), CSS (basic), Bash, Git

Languages: English, Korean