CARLOS PAREJA

(562)551-1298
cpareja3025@gmail.com

LinkedIn, GitHub, Personal Website

EDUCATION

University of California, San Diego *(expected) March 2024*
B.S. in Mathematics-Computer Science with a specialization in Machine Learning

Pasadena City College *2018-2021*
Associates in Physics, Mathematics, Natural Sciences

EXPERIENCE

Apple Cupertino, CA
Machine Learning Engineer Intern *June 2023 - September 2023*

- Lead the end-to-end development of a novel Large Language Model (LLM) for Apple's automated conversational assistant. My novel LLM has the capability to leverage real-time API data integrated with natural language capabilities.
- Designed and implemented the entire model-agnostic training pipeline of my novel LLM in preparation for deploying the LLM in a production setting.
- Fine-tuned multiple LLMs and utilized various Prompt Engineering techniques such as Few-Shot Prompting in order to develop a superior automated conversational assistant with my novel LLM.
- Utilized various Machine Learning technology stacks such as PyTorch, HuggingFace, AWS, DeepSpeed, Docker, and CUDA during the training and testing of my models. Overall, my novel LLM will impact 2.3 million Apple customers daily and will impact 80% of Apple customer-advisor conversation flows.

UC San Diego Duarte Lab La Jolla, CA
Machine Learning Research Intern *January 2022 - Present*

- Working on Computer Vision and High Energy Physics (HEP) research with Self-Supervised Learning (SSL) of Image Representations with the VICReg method. Developed a Convolutional Neural Network with the VICReg method that can perform downstream tasks such as Jet Classification, reaching similar results as supervised learning tasks.
- Developing the JetNet Python library for Machine Learning (ML) and HEP. JetNet is a library for developing and reproducing jet-based ML projects. We're currently averaging 2000 downloads per month. Published paper to the Journal of Open Source Software (JOSS). Presented JetNet at the AI and ML Session of the 2023 Conference on Computing in High Energy and Nuclear Physics (CHEP)
- Expanding JetNet by adding new simulated particle collision datasets and reformatting to PyTorch Tensors and NumPy arrays. Automating the ML data preprocessing task by taking a user's request for data and returning the HEP data that's ready to be trained on an ML model.

NASA Jet Propulsion Laboratory (JPL) Pasadena,CA
Software Engineer Intern *April 2022 - September 2022*

- I implemented and tested Markov Chain Monte Carlo (MCMC) sampling algorithms to facilitate in the research and analysis of data from upcoming galaxy redshift survey telescopes such as the Nancy Grace Roman Space Telescope and SPHEREx. The MCMC algorithms that I implemented from scratch and

utilized were able to make a tight constraint on cosmological parameters in a highly dimensional parameter space.

- Overall, writing code for two large-scale NASA missions allowed me to assist the research division of JPL and the parameters I efficiently constrained will give us further insight into the physics that were play at the beginning of the Universe.
- I developed my MCMC algorithms in the Julia and Python languages and worked with GitHub to collaborate with my colleagues and tested my MCMC algorithms on the TACC Supercomputer.

UC Santa Cruz Lamat (NSF REU) Santa Cruz, CA
Computational Astrophysics Researcher *April 2021-January 2022*

- I worked on high-order Gaussian Process methods for Computational Fluid Dynamics (CFD) simulations with the GP research group at UCSC.
- My research consisted of comparing and implementing three traditional finite differencing methods against the new GP modeling method. Our current work consists of expanding to include more numerical methods and I am implementing these new methods as well.
- I conducted my data analysis in Python, calculated the error for each method, and conducted a grid convergence study where I analyzed how error decreased when increasing the grid resolution of each numerical approximation. I demonstrated that with GP modeling we can achieve a highly accurate numerical approximation for CFD simulations.
- My research has wide-ranging applications, most notably in improving numerical accuracy in larger-scale CFD simulations in astrophysics and other fields of science and engineering.

PROJECTS

Implementing the Self-Supervised Learning Method, VICReg, and training on the MNIST dataset for classification May 2023

· The VICReg method consists of an encoder-decoder architecture. I utilized a CNN as the encoder and fully connected layers as the decoder. I implemented the VICReg method from scratch using PyTorch. VICReg model is fed two of the same images but one image is augmented with augmentations such as crop, color jitter, rotation, etc. The model was trained to minimize the distance between pairs of embeddings corresponding to the same image. I then fine-tuned my Pre-Trained VICReg model on the classification downstream task for the MNIST dataset. My VICReg Model achieved the same accuracy of 95% as a supervised task with labels on this classification task.

Implementing a Convolutional Neural Network (CNN) with Inception Modules for Neutrino Interaction Classification in PyTorch March 2023

· I implemented a CNN containing Inception Modules and trained the CNN on a simulated dataset of neutrino events. The neutrino event data contained data on two different views (X and Y) therefore I had two CNNs running in parallel and concatenated their respective outputs with an Inception Module I implemented from scratch using PyTorch. The dataset contained five different events therefore a random sample would give us a 20% accuracy. My classifier was able to achieve a validation accuracy of 64.2%.

Implementing a CNN to an LSTM model for Image Captioning of the COCO Dataset and ResNet50 to LSTM model November 2022

· I implemented a Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) network in order to generate captions for images of the COCO dataset. I also performed transfer learning by replacing my CNN with a ResNet50. The best model was with ResNet50 and I received a BLEU1 score of 54.5.

Implementing a Multi-Layer Neural Network from scratch for Multi-Class Classification of the CIFAR-10 Dataset October 2022

· I implemented a Multi-Layer Neural Network from scratch, without the use of any deep learning frameworks. I only used the Python NumPy library to construct the layers, implement back-propagation, and several optimizing algorithms from scratch such as Stochastic Gradient Descent (SGD) with Momentum and the Adam optimizer. I performed Multi-class classification on the CIFAR-10 Dataset with a Softmax output layer and our best model achieved an accuracy of 61%.

RELEVANT COURSES

1. Computer Science/Data Science: Data Structures and Algorithms (Java), Mathematical Software, Data Science in Practice, Deep Learning, Systems Programming, Supervised Machine Learning Algorithms, Machine Learning in Physics, Intro to Computer Science I (C++), Advanced Data Structures (C++), Design and Analysis of Algorithms, Web Mining and Recommender Systems

2. Mathematics:
Calculus I, Calculus II, Honors Multi-variable Calculus, Linear Algebra, Honors Differential Equations, Mathematical Reasoning, Statistical Methods, Discrete Math and Graph Theory, Modern Algebra, Optimization Methods for Data Science

SKILLS

Programming Languages:
Python, Java, Julia, R, C++, C, LaTeX, Matlab, SQL, x86 Assembly, ARM Assembly, HLA

Tools/Frameworks:
Git, Vim, Linux/UNIX Command Line, Jupyter, VsCode, Eclipse, R Studio, Slack, NumPy, Pandas, Matplotlib, Seaborn, TensorFlow, PyTorch, JUnit, Zeus, Scikit-Learn, AWS, HuggingFace, DeepSpeed

Languages:
English, Spanish

Soft Skills:
Customer Service, Teamwork, Leadership, Time Management, Communication, Problem Solving, Accountability