Jeremiah O. Dibble

Los Angeles, CA 90266 | 424-216-1785
jeremiah-dibble.github.io/ | LinkedIn.com/in/jeremiah-dibble | *Jeremiah.Dibble@gmail.com*

Work Experience

Machine Learning Engineer – Caltech Thomson Lab/OTTCP *2021 – Present*

- Implemented and tested custom loss functions to combine transformer models to add an unlimited number of additional tasks or contexts with significantly less additional training than traditional training. This led to 50%+ saving on compute costs because of the shorter training time.
- Improved a model sparsity algorithm using Pytorch to work on a more general set of NLP model architectures and automated the sparsification at a variety of different levels based on optimizing for a given target sparsity. Researchers in the lab leveraged these tools for data collection used to publish papers.
- Automated supervised and unsupervised fine-tuning of BERT or GPT NLP models on any given text dataset, to allow for the mass fine-tuning of different models and datasets.
- Leveraged Hugging Face to automate fine-tuning patient records to classify associated diagnosis codes and return embedded records based on the selected model. 97% accuracy was achieved with a relatively small BERT model.
- Preprocessed Waymo traffic point cloud into different formats to optimize the self-organizing neural net's (SNN) predictions of motion and interaction of multiple objects to determine the market viability of the SNN.

Machine Learning Engineer – Kapital Call *2020 – 2021*

- Developed and tested a variety of ML models to predict the future cashflows for illiquid private equity funds and used the fund-based results to predict the necessary cash reserves for a portfolio manager with 94.6% accuracy.
- Using a modified Deep Profit model and public market data we achieved ~95% accuracy for a 3-month prediction on the portfolio bases rather than on the fund level.

Electrical Engineer Intern – NASA Jet Propulsion Laboratory *2020 – 2020*

- Designed FPGA design flow software using Python that integrated into the JPL verification process to identify clockdomain crosses within an FPGA design and modify related files to eliminate extraneous errors during simulations.
- Automated the management of FPGA warnings, within JPL's Linux environment for FPGA verification, to significantly reduce the time engineers spend identifying and fixing low-priority errors.

Electrical Engineer Intern – Caltech Optical Observatories: Development *2019 – 2019*

- Designed and implemented a data processing system and corresponding hardware that increased the efficiency of a detector on the Thirty Meter Telescope International Observatory by 38%.
- Designed a PCB using Altium to interface between a custom bus and an FPGA.
- Programmed an FPGA in Verilog using Vivado to store data and serialize detector instructions.

Undergraduate Researcher – The Kirschvink Lab & The Force Group, Caltech *2017 & 2018*

- Designed and manufactured an antenna array in Solid Works to produce a localized RF field.
- Redesigned a water tunnel experiment to increase the data collection rate by a factor of three, without deterioratingthe quality of the turbulence data.

Education – *California Institute of Technology* *Graduated 2021*

- B.S. in Electrical Engineering and
- B.S. in Business, Economics, and Management

Projects *2018 – 2020*

Programming Projects (Python)

- Developed a dynamic neural architecture search framework that allows for a fully connected single hidden layer neural network to be dynamically resized using an RL controller to achieve the ideal number of parameters.
- Used decision tree gradient boosting to predict the cause of fires with 83% accuracy based on location, time, and size of the fire. Performance was significantly improved by preprocessing the data and optimizing regularization.
- Wrote a singular value decomposition algorithm which was trained using stochastic gradient descent. This algorithm was used to create a model to predict movie ratings within one star out of five and visualize recommendations.
- Created a benchmarking program, in Python, to compare one's trading history to a fixed portfolio with identical trade time and investment to determine if, time-weighted, one is outperforming a fixed portfolio of one's choice.