Xiongjie (Jack) Dai

+86-133-8052-0460 | xiongjie.dai.1@gmail.com | linkedin.com/in/xiongjie-dai | github.com/Xiongjiedai

Education

University of Illinois Urbana-Champaign (UIUC) — Urbana-Champaign, Illinois, USA
Master of Science in Statistics; ***GPA: 3.84/4.00*** — *August 2021 – May 2023*

Jinan University / University of Birmingham — Guangzhou, China/Birmingham, England
Bachelor of Economics ***GPA: 3.93/4.25*** — *September 2017 – June 2021*
Bachelor of Science in Applied Mathematics with Statistics (First-class degree)

Skills

Machine Learning Research: Artificial Intelligence, Deep Learning, Natural Language Processing, Statistical Learning
Data Science: Data Science Programming Methods, Statistical Data Management, Statistical Consulting
Programming: Python, R, SQL, Computer Science
Technologies: Git, AWS, Shell, PyTorch, TensorFlow, NumPy, Pandas, Matplotlib, ggplot2, Tidyverse, Gradio, Docker

Work Experience

Artificial Intelligence / Machine Learning Engineer — Wisping LLC, Illinois
Git*, Shell, RunPod, LLM, GenAI* — *July 2023 – July 2024*

- Engineered comprehensive **LLaMA** model tests, comparing NVIDIA GPUs to Apple Silicon using the **llama.cpp** toolkit, adapting an innovative cloud-based strategy for performance evaluation to establish proof of concept for local Large Language Models (LLMs) on devices; garnered industry accolades evidenced by over **1.2k** Github stars.
- Documented and analyzed inference speeds for various model sizes across different quantizations and multiple hardware configurations, providing actionable insights on the optimal setups for efficient LLM inference.

Graduate Course Assistant — University of Illinois Urbana-Champaign
Python, R, SQL, Deep Learning, Statistical Learning — *September 2022 – December 2022*

- Collaborated with faculty to devise engaging machine learning course content and innovative coding assessments in R and Python; Graded students, and resolved academic issues during weekly office hours, with a **0** course-complaint rate.

Projects

Neural Machine Translation: RNN and Transformer Models | *Python, PyTorch* — *May 2023*

- Led the development of a Neural Machine Translation (NMT) system, leveraging PyTorch to architect both Recurrent Neural Network (RNN) and **Transformer** models, focusing on the translation of Spanish to English text.
- Designed and implemented the RNN using Gated Recurrent Units (GRUs) and attention mechanisms, constructing encoder and decoder components. Enhanced translation quality in the Transformer model with multi-head attention and positional encoding to capture sentence structure and preserve word order.
- Achieved BLEU-4 scores of 0.058 for the RNN model and 0.059 for the Transformer model.

Reinforcement Learning in Snake | *Python, Numpy* — *December 2022*

- Engineered a reinforcement learning AI agent to master the Snake game, employing a Temporal-Difference (TD) **Q-learning** algorithm in a defined Markov Decision Process (MDP) framework of states, actions, and rewards.
- Implemented an exploration policy to balance between exploring new states and exploiting known ones, ensuring comprehensive learning coverage, and used a decaying learning rate to optimize the Q-value update process over time.
- Analyzed agent's performance through rigorous testing phases to ensure robustness and adaptability of the model.

Sentiment Analysis for Amazon Review & Drug Dataset | *R markdown, Word2Vec* — *December 2022*

- Advanced sentiment analysis on Amazon and Drug Review datasets by and comparing four classic embedding and **NLP** methods (BoW, Word2Vec, GloVe, fastText), achieving best performance with FastText (86.49% accuracy on Amazon, 78.69% on Drug), demonstrating FastText's ability to handle unseen words by utilizing subword information.
- Employed data preprocessing techniques for text normalization and vectorization. Filtered out common words using two-simple t-tests to interpret Word Cloud. Leveraged Naive Bayes and Random Forest algorithms for classification.

Explaining the Effects of Data Augmentation with CNN | *Python, PyTorch, Matplotlib, Gradio* — *May 2022*

- Developed and evaluated three distinct Convolutional Neural Network (CNN) models across various seeds to determine the impact of augmentation techniques on model performance, focusing on a binary cat-dog classification challenge.
- Applied regularization (dropout, L1, L2) to avoid overfitting, with binary cross entropy as cost function and SGD as optimizer; using flipping data augmentation, best CNN model showed accuracy improvement from 59.6% to 74.8%.