Xiongjie (Jack) Dai

☎ 217-305-0416 | @ xdai12@illinois.edu | 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 (JNU) Guangzhou, China
Bachelor of Economics ***GPA: 3.93/4.25*** *September 2017 – June 2021*

University of Birmingham (UoB) Birmingham, England
Bachelor of Science in Applied Mathematics with Statistics (First-class degree) *September 2017 – June 2021*

Relevant Coursework & Skills

Machine Learning: Artificial Intelligence, Deep Learning, Natural Language Processing, Statistical Learning
Data Science: Data Science Programming Methods, Statistical Data Management, Statistical Consulting
Programming: Python, R, SQL
Technologies: Git, Shell, PyTorch, Keras, NumPy, Pandas, Matplotlib, ggplot2, Tidyverse, R Shiny app

Projects

GPU Benchmarks on LLM Inference | *Git, Shell* *December 2023*

- Orchestrated comprehensive benchmarks of **LLaMA** models across a spectrum of NVIDIA GPUs and Apple Silicon devices, utilizing the state-of-the-art open-source LLM inference toolkit **llama.cpp** for in-depth performance analysis.
- Documented and analyzed inference speeds for various model sizes across different quantization and multiple hardware configurations, providing actionable insights on the optimal setups for efficient LLM inference **on device**.
- Achieved notable recognition within the open-source community, with the GitHub repository amassing over 200 stars.

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 model using Gated Recurrent Units (GRUs) and attention mechanisms to construct both encoder and decoder components. Employed a multi-head **attention** mechanism in the Transformer model to enhance translation quality by capturing different aspects of sentence structure and integrated positional encoding within the Transformer model to preserve word order information.

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, consisting of states, actions, and rewards, to navigate a snake through a grid environment towards food pellets.
- 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* *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 techniques, such as dropout, L1, and L2 penalties, to avoid overfitting. Binary cross entropy was used as the cost function, and SGD was used as the optimizer. The best CNN model achieved accuracy with 74.8% under the data augmentation technique of flipping, as compared with 59.6% from the original data.

Recommendation System for Movie | *R, Tidyverse, R Shiny app* *December 2021*

- Constructed two recommendation systems: one leveraging genre preferences to suggest popular films and another employing collaborative filtering to recommend movies based on user ratings and item characteristics. Utilized Tidyverse suite for data manipulation and analysis, designed and built an interactive **shiny app** based on the system.

Work Experience

Graduate Course Assistant University of Illinois Urbana-Champaign
Fundamentals of Deep Learning, Statistical Learning *September 2022 – December 2022*

- Effectively collaborated with faculty and assisted students, developing machine learning content and creative coding assessments in R and Python. My role in grading and feedback highlights my strong **self-directed** problem-solving skills, communication ability, and passion for contributing constructively to team-driven machine-learning projects.