Xianglong Xu

Personal Website: sean652039.github.io

412-452-5278 sean652039@gmail.com/xix110@pitt.edu linkedin.com/in/xianglongxu github.com/Sean652039

Education

University of Pittsburgh **August 2023 – May 2025**
Master of Science in Information Science *GPA: 3.90/4.0*

North China Electric Power University **September 2019 – June 2023**
Bachelor of Science in Computer Science - Specialization in AI *GPA: 3.59/4.0*

Skills

Languages and Tools: Python, Bash, Git, Scikit-learn, Torch, PyTorch Lightning, Tensorflow/Keras, Pandas, Numpy, Matplotlib, Seaborn, Scipy, AWS, Unix, Pytest, SQL, R, MATLAB, Java, HTML, CSS, JavaScript
Machine Learning and Algorithms: Linear and Logistic Regression, Neural Networks, Boltzmann Machines, Variational Autoencoders, Stable Diffusion, Transformer, Reinforcement Learning, Generative Adversarial Networks

Experience

University of Pittsburgh **November 2024 – May 2025**
Research Assistant – NLP Research *Pittsburgh, Pennsylvania*

- X. Xu, J. Bowen, and R. Taheri. *Token Masking Improves Transformer-based Text Classification*, 2025.
- Designed a lightweight token masking mechanism that introduces controlled input perturbations and uses gradient averaging across training steps as a regularization method to encourage more generalized representations.
- Developed and validated a forward token-masking framework with adaptive probability tuning ($p = 0.0$–0.5) for transformer-based text classification, evaluated on multiple tasks in the LinCE multilingual benchmark suite.
- Conducted experiments on language identification and sentiment analysis with diverse transformer models (e.g., Qwen, TinyLlama), finding 0.1 to be an optimal masking rate, with variation across tasks and model sizes.

Vosyn Inc. **May 2024 – August 2024**
AI Software Developer Intern *Chicago, Illinois*

- Designed and implemented an end-to-end Python-based Speaker Diarization System to gradually replace the pyannote library due to its long runtime, enabling faster identification of multiple speakers' timelines in audio.
- Implemented a Chinese Restaurant Process-based approach in the clustering, achieving 10x faster execution time than pyannote, and trained a Speaker Change Detection model, enabling more accurate speaker timelines.
- Integrated WebRTC's Voice Activity Detection with a ring buffer to reduce detection errors by considering contextual frames and employed temporary files with io.BytesIO for large audio files to minimize memory usage.
- Developed evaluation using the AMI Meeting Corpus datasets to calculate Diarization Error Rate, providing a metric for developers to assess the improvements made to the system, enabling continuous optimization.

Projects

Music Generation with Deep Learning **September 2024 – December 2024**

- Designed and implemented a dual-model music generation system that leverages Stable Diffusion and Restricted Boltzmann Machines for generating roll plots from music MIDI files, which are then converted to MIDI files.
- Preprocessed plots for training by truncating the x-axis representing time, expanding the y-axis (music pitch range) by replicating pitch rows to match input dimensions, and applying color-coding to increase dimensions.
- Designed a Stable Diffusion training pipeline using a UNet2DModel with adaptive pixel-wise weighted loss, dynamic noise scheduling, and checkpoint management, with architecture aligned to the preprocessing of plots.
- Designed an RBM for denoising, treating each pixel as a visible node with a hidden layer of half the visible nodes, while incorporating adaptive noise generation, multi-type noise injection, and contrastive divergence training.
- Engineered an image-to-MIDI conversion pipeline that transforms generated plots into binary representations through adaptive thresholding, translating pixel data into MIDI note sequences with time and pitch mapping.

AI Snake **March 2024 – May 2024**

- Optimized a reinforcement learning AI agent for the 'Snake' game using Proximal Policy Optimization with MLP and CNN architectures, enhancing the reward-punishment mechanism and achieving a 5% score improvement.
- Engineered a Markov chain model to analyze the game process by treating each time step as a Markov state, suggesting that compressing the snake's body to reduce the space it occupies is one potential optimization.
- Developed a recursive territory calculation algorithm inspired by Go game rules to compute the empty cells enclosed by the snake's body, using the number of enclosed cells as a penalty criterion to train the agent.