Jingcheng Wu

(412) 500-1834 | jingchew@cs.cmu.edu | www.linkedin.com/in/wu-jingcheng | www.github.com/Wu-Jingcheng | wu-jingcheng.github.io

EDUCATION

Carnegie Mellon University, **School of Computer Science** — *Pittsburgh, PA*

Master of Science in Artificial Intelligence & Innovation | GPA: 4.0/4.0 — *08/2021-05/2023*

University of Electronic Science & Technology of China — *Chengdu, CN*

Bachelor of Engineering in Electronic Information Engineering | GPA: 3.93/4.0 — *09/2016-07/2020*

SKILLS

Languages and Libraries: Python, Java, C/C++, JavaScript, PyTorch, NumPy, Pandas, Ray[RLlib], Gym, CVXPY, Markdown

Tools: VSCode, Azure, Docker, Jupyter, Git, GDB, Django, Spring Boot, Thymeleaf, Shell, LATEX, MySQL, Redis

Realms: Machine Learning, Object-Oriented Design, Backend & Full Stack Development, Operations Research

WORK EXPERIENCE

Microsoft – Machine Learning Group — *Beijing, CN*

Research and Software Development Intern — *09/2020-08/2021*

Linear Programming & Supply Chains Simulator Development **[Python, CVXPY, Docker]**

- Prototyped base-stock policies and its three modes with CVXPY using Python, providing operations research (OR) based implementations to solve inventory management problem for agents in supply chains.
- Contributed to MARO in Python by building a supply chain simulator module, and earned ~500 stars on GitHub.

Multi-agent Reinforcement Learning in Replenishment Problem **[Python, PyTorch, Ray, RLlib]**

- Designed a centralized-training-decentralized-execution multiagent reinforcement learning strategy using shared parameter policies with Ray and RLlib, providing scalable solutions to inventory management problem.
- Applied context augmentation and contextual predictive auxiliary tasks to alleviate the overfitting issue in reinforcement learning to inventory management problem with PyTorch by proposing Contextualized MDP.
- Obtained better performance in 70% of settings in comparison to base-stock policies and standard DQN.

Adaptive Truncation Bootstrapping for On-policy Reinforcement Learning **[Python, Azure, Docker, Gym]**

- Proposed adaptive truncation bootstrapping for on-policy learning and improved performance on 100% Atari games with Gym and PyTorch, and solved Mountain Car with 70% shorter episodes.
- Ran relative experiments with Azure and Docker by building containers pulled from a self-built uploaded image.

PROJECTS

Carnegie Mellon University — *Pittsburgh, PA | Remote*

Cache Simulator **[C]** — *05/2020-08/2020*

- Developed a cache simulator using C, customized by user specified associativity, block size and number of sets, by maintaining a tabular array with set index as row and recency as column to simulate LRU eviction policy.
- Designed matrix transpose algorithms to achieve significant speedup with strategies including blocking to fit in cache design of given parameters, saving 72% computational clock cycles compared to simple row-wise scan transpose.

Heap Memory Allocator **[C]** — *05/2020-08/2020*

- Initiated a memory allocator and designed its free, malloc, realloc and calloc functions with segregated list, footerless allocated blocks, better fit algorithms, adjacent free blocks coalescing and smaller minimum block size.
- Achieved 109% throughputs and 74% utilizations of reference design performance in average on reference traces.

Proxy Web Server **[C, multithreading]** — *05/2020-08/2020*

- Created a proxy server in C that is able to concurrently respond to requests with multithreading and cache response objects, where the caching mechanism is designed to implement an LRU eviction policy based on double linked lists.

Tiny Shell Program **[C, multiprocessing]** — *05/2020-08/2020*

- Prototyped an interactive command-line shell in C utilizing multiprocessing to support running executable programs in foreground/background, 4 builtin commands and handling various interrupts.

University of Electronic Science and Technology of China — *Chengdu, CN*

Online Bookstore Website **[Java, JavaScript, Spring Boot, HTML, CSS, MySQL]** — *02/2019-07/2019*

- Built an online bookstore website using Java on Heroku and its modules including user signup and login, profile management, product management, shopping cart, order checkout and history, and automatic email confirmation.

Adversarial Attack on Object Classifiers **[Python, Caffe]** — *09/2019-05/2020*

- Implemented FGSM algorithms to fool an image classifier VGGnet and accomplished over 90% fooling rate among more than 100 object classes and more than 200 images from ImageNet.