

ZHIWEI LIU

✉ zhiwei.liu@campus.tu-berlin.de · ☎ (+49) 015781447744 · in Zhiwei Liu

🎓 EDUCATION

Technical University of Berlin, Berlin, Germany 2017 – 2019

Master student in Computer Science with Cognitive Systems as major

Xidian University, Xi'An, China 2012 – 2016

B.S. in Software Engineering

👤 EXPERIENCE

DAI-Labor Berlin, Germany April 2018 – Present

Student Research Assistant IRML Department

- Implementing machine learning algorithms (e.g., MLP, CNN, RNN, and some state-of-the-art algorithms from latest research papers) based on Netlib interfaces (mainly BLAS and LAPACK) and Java bindings of CUDA library (a.k.a., JCuda, and I fixed a bug for this project as an open source contributor)
- Optimizing implemented algorithms from both mathematic and engineering sides (e.g., I have proposed a method to replace matrix-vector multiplication with matrix-matrix multiplication in certain cases, and it speeds up more than 10 times faster than the previous version, especially in large matrices computation)
- Research on parameter-free training and online optimization problems, currently focusing on searching for approximate global minimum for non-convex problems with no prior information (e.g., hyper parameters, diameter of parameters, gradient bounds of loss functions, etc.)
- Benchmarking the implemented algorithms with Tensorflow and/or Spark, guaranteeing faster convergence and/or higher performance

Fraunhofer FOKUS Berlin, Germany Feb. 2017 – Feb. 2018

Student Research Assistant NGN Department

- Responsible for implementation of the benchmarking tool component of 5G core network project
- Implemented key metrics tracking and scenario session management.
- Designed and implemented automatic scenario generation algorithm such that batch of complex test cases could be generated in an easier way.

👤 PROJECTS

DSLs for Scalable Data Analysis May 2018 – August 2018

C++ Individual Project

- By using LLVM & Clang framework and library, I implemented a compiler of SQL written in C++, and its idea is from the paper Emma
- With the compiler, programmers do not have to recite the SQL (or Spark, etc.) APIs. Instead, programmers only need to express the SQL statements in a normal logic written in C++. The compiler will compile it to lower level representations which is equal to corresponding SQL statements
- Evaluated and benchmarked on CoGaDB (Column-oriented GPU-accelerated DBMS) using TPC-H database

⚙️ SKILLS

- **Programming Languages:** C, Python, Java, C++
- **Development:** scikit-learn, TensorFlow, PyTorch, Keras, CUDA, LLVM, MySQL
- **Platforms & Tools:** Linux, Docker, JupyterLab, GitLab, GitHub, SVN, tmux
- **Personal Blog:** liuzhiwei.me