Jiahang Li

Beijing University of Posts and Telecommunications ljh1064126026@bupt.edu.cn

EDUCATION

Beijing University of Posts and Telecommunications Sep 2018 - Jun 2022

Data Science and Big Data Technique in School of Computing Beijing

- Bachelor degree of Engineering
- GPA: 3.56/4.0, 86.37/100
- IELTS: 6.5
- Correlated curriculums: Linear Algebra, Matrix Theory, Calculus, Discrete Mathematics, Principles of Database System, NoSQL Database Theory, Operating System, Introduction to Computer Systems, Machine Learning, Data Mining, etc

HONORS & AWARDS

Second Prize of National English Competition for College Students Mar 22th, 2021

Third Prize of National Mathematics Competition for College Students Apr 2nd, 2021

Honorable Mention of MCM/ICM Apr 30th, 2021

RESEARCH INTEREST

I'm greatly interested in large-scale (GNN) (distributed) system, graph representation learning and its application, high performance computing and its combination with machine/deep learning.

SKILLS LIST

- **Computer Languages**: C/C++/Python/SQL/Linux shell script
- **Tools**:
  - Machine and Deep learning frameworks including NumPy/Sklearn/PyTorch/DGL/Optuna, etc.
  - HPC tools including CUDA C/Cython, etc.
  - Have some knowledge of NoSQL and distributed systems including HBASE, HDFS, Hadoop, Spark, Horovod, etc.
- **Hobbies**: playing basketball, reading books, watching movies, mathematics

PUBLICATION

Xiang Song, Runjie Ma, **Jiahang Li**, Muhan Zhang, David Paul Wipf, **Network in Graph Neural Network** [pdf]

RESEARCH AND PROJECTS EXPERIENCE

OpenHGNN Feb 2021 - Present

Research Assistant (GAMMA Lab affiliated with BUPT) Beijing

OpenHGNN: https://github.com/BUPT-GAMMA/OpenHGNN

GAMMA Lab: https://github.com/BUPT-GAMMA

OpenHGNN, which is aimed at building an open-source Heterogeneous GNN framework based on DGL and Pytorch, is one of the key projects of **GAMMA Lab** directed by **Prof. Chuan Shi**.

Reproduced MAGNN, one of the Heterogeneous GNN SOTA model:

- Several trials have been carried out to show that **my implementation has the performance at least 2% higher than that of the author on IMDB for all dataset partition settings.**
- **Reduced code complexity.** For example, the model nesting level has been reduced from 4 to 2, so that user would find it easier to track the logic of codes.
- **Converted datasets used in MAGNN into DGL graphs**, which avoids users processing the datasets on their own and allows them to make use of datasets easily.

Reproduced MAGNN sampler for scaling MAGNN to mini-batch training on large graphs:

- **Introduced natural join of database theory to efficiently sample meta-path-based neighbors**, which simplifies the redundant codes and complicated logic of the original implementation introduced by the author, and reduce time overhead of MAGNN sampler.
- **Generalized MAGNN sampler to any HGNN meta-path-based task with DGL graphs.** The original implementation has been limited to specific datasets and tasks of MAGNN. My generalization allows users to easily employ the sampler in their own tasks and datasets.
- **Generalized MAGNN sampler to multi-layer sampling.** The original implementation only supports 1-layer sampling because the time overhead of sampling multi-layer neighboring nodes will be much high. My implementation reduces time and memory overhead of the sampler thus it can be generalized to multi-layer sampling.

Deep Graph Library (DGL) Jul 2021 - Present

Software Development Engineer Intern (DGL team affiliated with AWS Shanghai AI Lab) Shanghai

[DGL: https://github.com/dmlc/dgl](https://github.com/dmlc/dgl)

[AWS Shanghai AI Lab: https://www.amazonaws.cn/en/ailab/](https://www.amazonaws.cn/en/ailab/)

DGL, which is aimed at building an open-source GNN framework based on Pytorch, Tensorflow, and Mxnet, is one of the key frameworks of **AWS Shanghai AI Lab** directed by **Prof. Zheng Zhang.**

Reproduced GraphSAINT, one of the GNN graph-based sampling methods:

- **Achieved better performance on some datasets,** including 2%~4% higher F1-macro on Amazon dataset, at least 1% higher F1-macro on PPI dataset of the node sampler compared with the author's implementation.
- **Reduced sampling time overhead on some datasets.** For example, the reproduced random walk sampler has the sampling time less than **half** of that of the original author's implementation on the largest dataset used in GraphSAINT, that is, the Amazon.
- **Employed multiprocess parallelism of torch.DataLoader to speed up the sampling.** I replace codes of original author with torch.DataLoader to **reduce the complexity.** Also, Cython has been used to accelerate sampling by multithread mechanism, of which the reduction of time overhead is not significant. Thus Cython was discarded and implementation based on pure C was utilized.
- **Provided more sampling options.** The author only offers offline sampling, but I implement online sampling and offline sampling, so that users can choose how to trade off sampling time and model performance by choosing different sampling methods.
- **Simplified the codes and provided more annotations so that users can easily understand and use it.**

Working on the research project Network in Graph Neural Network with me as the third author:

- **Responsible for testing time and gpu memory overhead,** including testing the ratio between the training time of each epoch and the number of parameters, of which the result shows that for the same amount of parameters, our proposed model has less training time of each epoch than the baselines.
- **Working on a new research project** under the supervision of **Dr. David Wipf** and **Dr. Xiang Song**, which is still in the progress. The project is based on the Network in Graph Neural Network.