ACCELERATED GNN TRAINING WITH DGL AND RAPIDS CUGRAPH IN A FRAUD DETECTION WORKFLOW

KDD ‘22, WASHINGTON DC,

AUG 14TH , 2022

ACCELERATED GNN TRAINING WITH DGL/PYG AND RAPIDS CUGRAPH IN A FRAUD DETECTION WORKFLOW

KDD ‘22, WASHINGTON DC,

AUG 14TH , 2022

SPEAKER INTRODUCTION

Brad Rees is a Senior Manager at NVIDIA and lead of the RAPIDS cuGraph team. Brad has been designing, implementing, and supporting a variety of advanced software and hardware systems for over 30 years. Brad specializes in complex analytic systems, primarily using graph analytic techniques for social and cyber network analysis. Brad has a Ph.D. in Computer Science from the Florida Institute of Technology.

Xiaoyun Wang is an applied scientist at NVIDIA Corp, cuGRAPH team. Xiaoyun earned her Ph.D. in computer science in 2020 and M.S. in statistics in 2015 from the University of California, Davis. Xiaoyun got her B.S. in applied mathematics from the Sun Yat-Sen University in 2013. Xiaoyun is interested in graph neural networks(GNNs), robustness in deep learning models and large-scale machine learning problems.

SUPPORT

Joe Eaton is the Distringuished System Engineer for Graph and Data Analytics at NVIDIA. He works on RAPIDS, dividing time between cuML and cuGRAPH. His interests are general optimization and applications of sparse linear algebra to industrial scale problems. Previously, he was manager for sparse linear algebra CUDA libraries cuSPARSE, cuSOLVER, and nvGRAPH, and managed AmgX, now an open source package of GPU-accelerated sparse iterative solvers. Joe lives in Austin, Texas, and holds a Ph.D. in computational and applied mathematics from UT Austin, a master's in mechanical engineering from Stanford, and a bachelor's in mechanical engineering from Rice University.

Rick Ratzel is the Lead for cuGraph Libraries. Rick joined NVIDIA just over three years ago, bringing several years of experience as a technical lead for teams in industries that include test and measurement, electronic design automation, and scientific computing. Rick’s focus for cuGraph, and throughout his career, has been on software architecture and API usability.

Onur Yilmaz is a senior deep learning software engineer who's been with NVIDIA for more than five years. He's one of the main engineers who contributed to RAPIDS cuML, the GPU-accelerated machine learning open source library, and he also contributed to Merlin, an open-source framework for building large-scale deep learning recommender systems. He is currently making graph neural network training and deployment easy for data scientists, researchers, and engineers. Onur holds a Ph.D. in computer engineering from the New Jersey Institute of Technology. His dissertation focused on traditional machine learning and high-performance computing for finance.

Dominique LaSalle has been at NVIDIA for three years, where he works on optimizing the tools used for Graph Neural Networks. Prior to this, he worked on iterative and direct solvers for physics simulations. He received his Ph.D. from the University of Minnesota in 2015, where his research focused on graph partitioning and sparse matrix re-ordering.

AGENDA

3 Sections with a break between sections

- Welcome
- Introduction to RAPIDS
  - cuDF for ETL
  - cuML for GBDT training
  - Workflow using xgboost

break

- cuGraph
  - Overview
  - Property Graphs
  - Graph-as-a-Service
  - Workflow – graph + xgboost
- Introduction to GNN
- Fraud Detection combining graph features and GNN embeddings
  - Workflow

break

- DGL + cuGraph Local
  - Intro to DGL
  - code sample
- PyG + GaaS
  - Intro to PyG
  - code sample
- Conclusion and Q&A

Please ask questions as we go.
Some questions might be held to the end of section

FOREWORD

- We work closely with both the DGL and the PyG teams
  - But what we are presenting is our work and might not represents DGL or PyG
- RAPIDS
  - https://rapids.ai/
- Deep Graph Library (DGL)
  - https://www.dgl.ai/
- PyG
  - https://www.pyg.org/

- Some of the examples will present are based on our local forks of DGL and/or PyG, but that code is slated to be pushed into the next releases

GOAL AND INTENT OF TUTORIAL

Problem Statement

- The goal of this tutorial is on presenting how to create GNN workflows using RAPIDS and either DGL or PyG
  - We are using fraud detection as and example, we are not fraud experts

- Fraud detection system have been developed for years, we want to be able to develop new AI-based solutions

- "*In law, **fraud** is intentional deception to secure unfair or unlawful gain, or to deprive a victim of a legal right.*" - Wikipidia

- LexisNexis *True Cost of Fraud*™ Study show
  - the cost of fraud continues to raise, now cost retailer $3.60 for every $1 of fraud
  - The amount of fraud is increasing,
  - https://risk.lexisnexis.com/insights-resources/research/us-ca-true-cost-of-fraud-study

THE DATA

- Dataset requirements
  - easy to get
  - fully sharable
  - size that allows people to test workflows on any size GPU

- Selection
  - Elliptic Data Set
    - https://www.kaggle.com/datasets/ellipticco/elliptic-data-set
    - Bitcoin and illicit transactions
    - The Graph is 203,769 verticves and 234,355 edges
    - 2% are labeled as illicit and 20% labeled as licit – so 78% are unlabeled
    - 166 featured per vertex

JUPYTER NOTEBOOKS

- The notebooks can be found at:

  https://github.com/rapidsai-community/event-notebooks/tree/main/KDD_2022

  just remember this part and you will be able to find notebooks

- The Notebook will NOT be run live
  - I never trust remote connections when doing a presentation
  - using PowerPoint since it is easy to annotate notebook

- My Test Hardware:
  - HP Z840 Workstation
  - Dual Intel(R) Xeon(R) CPU E5-2643 v4 @ 3.40GHz
  - 64GB RAM
  - NVIDIA RTX A6000