



Graph Neural Networks in TensorFlow

Bryan Perozzi
bperozzi@acm.org
Google Research
New York, USA

Sami Abu-El-Haija
haija@google.com
Google Research
Mountain View, USA

Anton Tsitsulin
tsitsulin@google.com
Google Research
New York, USA

ABSTRACT

Graphs are general data structures that can represent information from a variety of domains (social, biomedical, online transactions, and many more). Graph Neural Networks (GNNs) are quickly becoming the de-facto Machine Learning models for learning from Graph data and hereby infer missing information, such as, predicting labels of nodes or imputing missing edges.

In this tutorial we'll cover essential applications of Graph Machine Learning using TensorFlow GNN¹ [12], a Python framework that extends TensorFlow [1] with Graph Neural Networks (GNNs) [9]: models that leverage graph-structured data. TF-GNN is motivated and informed by years of applying graph representation learning to practical problems at Google [2–8, 10, 11, 13–19, 22–29]. In particular, TF-GNN focuses on the representation of *heterogeneous* graph data and supports the explicit modeling of an arbitrary number of relationship (edge) types between an arbitrary number of entity (node) types. These relationships can be used in combination with other TensorFlow components, e.g., a TF-GNN model might connect representations from a language model to those of a vision model and fine-tune these features for a node classification task. Many teams at Google run TF-GNN models in production. We believe this to be a direct consequence of TF-GNN's multi-layered API which is designed for accessibility to developers (regardless of their prior experience with machine learning).

ACM Reference Format:

Bryan Perozzi, Sami Abu-El-Haija, and Anton Tsitsulin. 2023. Graph Neural Networks in TensorFlow. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, August 6–10, 2023, Long Beach, CA, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3580305.3599177>

1 LEARNING GOALS

We designed this tutorial for industry practitioners and academic researchers, who would like to: 1) Apply ML techniques on graph data, without worrying about implementation and scaling details; and/or, 2) Extend the research field of ML on graphs, by starting from state-of-the-art models and graph learning techniques. The tutorial will include both conceptual details (mathematical derivations) and hands-on experience (running code in collaborative notebooks).

¹<https://github.com/tensorflow/gnn>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

KDD '23, August 6–10, 2023, Long Beach, CA, USA

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0103-0/23/08.

<https://doi.org/10.1145/3580305.3599177>

2 TUTORIAL CONTRIBUTORS

- **Bryan Perozzi**, *Google Research*. Bryan has been working on graph-based machine learning for over 10 years and has 14,000+ citations in the area. Bryan completed his Ph.D. at Stony Brook University, and his thesis [20, 21] won the 2017 ACM SIGKDD award for best dissertation. Bryan started the Graph Neural Network group inside Google Research and has worked on over 100 applications of graph-based learning at the company.
- **Sami Abu-el-Haija**, *Google Research*. Sami completed his Ph.D. at the University of Southern California, focusing on efficient training of Graph Neural Networks. His research contributions are on his Google Scholar.
- **Anton Tsitsulin**, *Google Research*. Anton is a researcher at Google working on TF-GNN and unsupervised graph embedding. He has published papers in the area at KDD, NeurIPS, ICLR, WWW, and VLDB.

3 TUTORIAL OUTLINE

A brief overview of the schedule is as follows:

- (1) **Section 1: Background**. We begin by covering some introductory material about GNNs and graph-based learning. **Application: Node Classification**. The application for this section is predicting a label for individual nodes in a graph (our most popular application).
- (2) **Section 2: Advanced Modeling**. The next section covers details of TF-GNN's model building API that allows stacking convolutions. **Application: Link Prediction**. In this application we'll discuss how to build models that can read out different parts of the graph structure and rank the likelihood that two nodes should be connected.
- (3) **Section 3: TF-GNN Tooling**. Finally, we'll cover some of the tooling designed to make it easy to build TF-GNN models. **Application: Unsupervised Embedding**. Here we'll cover using unsupervised learning in a GNN to learn features that can be re-purposed in other (non-graph) models.

4 MATERIAL

Slides, notebooks, and other material will be available at the tutorial website².

²https://github.com/tensorflow/gnn/blob/main/examples/tutorials/kdd_2023/README.md

REFERENCES

- [1] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 265–283.
- [2] Sami Abu-El-Haija, Amol Kapoor, Bryan Perozzi, and Joonseok Lee. 2020. Ngcn: Multi-scale graph convolution for semi-supervised node classification. In *uncertainty in artificial intelligence*. PMLR, 841–851.
- [3] Sami Abu-El-Haija, Bryan Perozzi, and Rami Al-Rfou. 2017. Learning edge representations via low-rank asymmetric projections. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1787–1796.
- [4] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A Alemi. 2018. Watch your step: Learning node embeddings via graph attention. *Advances in neural information processing systems* 31 (2018).
- [5] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*. PMLR, 21–29.
- [6] Rami Al-Rfou, Bryan Perozzi, and Dustin Zelle. 2019. Ddgc: Learning graph representations for deep divergence graph kernels. In *The World Wide Web Conference*. 37–48.
- [7] Aleksandar Bojchevski, Johannes Klicpera, Bryan Perozzi, Amol Kapoor, Martin Blais, Benedek Rózemerczki, Michal Lukasik, and Stephan Günnemann. 2020. Scaling graph neural networks with approximate pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2464–2473.
- [8] Kaidi Cao, Phitchaya Mangpo Phothilimthana, Sami Abu-El-Haija, Dustin Zelle, Yanqi Zhou, Charith Mendis, Jure Leskovec, and Bryan Perozzi. 2023. Learning Large Graph Property Prediction via Graph Segment Training. *arXiv preprint arXiv:2305.12322* (2023).
- [9] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Re, and Kevin Murphy. 2022. Machine Learning on Graphs: A Model and Comprehensive Taxonomy. *Journal of Machine Learning Research* 23, 89 (2022), 1–64.
- [10] Alessandro Epasto, Vahab Mirrokni, Bryan Perozzi, Anton Tsitsulin, and Peilin Zhong. 2022. Differentially Private Graph Learning via Sensitivity-Bounded Personalized PageRank. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). <https://openreview.net/forum?id=Fhty8PgFkDo>
- [11] Alessandro Epasto and Bryan Perozzi. 2019. Is a single embedding enough? learning node representations that capture multiple social contexts. In *The world wide web conference*. 394–404.
- [12] Olexandr Ferludin, Arno Eigenwillig, Martin Blais, Dustin Zelle, Jan Pfeifer, Alvaro Sanchez-Gonzalez, Sibon Li, Sami Abu-El-Haija, Peter Battaglia, Neslihan Bulut, et al. 2022. TF-GNN: graph neural networks in TensorFlow. *arXiv preprint arXiv:2207.03522* (2022).
- [13] Kimon Fountoulakis, Dake He, Silvio Lattanzi, Bryan Perozzi, Anton Tsitsulin, and Shenghao Yang. 2022. On Classification Thresholds for Graph Attention with Edge Features. *arXiv preprint arXiv:2210.10014* (2022).
- [14] Jonathan Halcrow, Alexandru Mosoi, Sam Ruth, and Bryan Perozzi. 2020. Grale: Designing Networks for Graph Learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Virtual Event, CA, USA) (KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2523–2532. <https://doi.org/10.1145/3394486.3403302>
- [15] Amol Kapoor, Xue Ben, Luyang Liu, Bryan Perozzi, Matt Barnes, Martin Blais, and Shawn O'Banion. 2020. Examining covid-19 forecasting using spatio-temporal graph neural networks. *arXiv preprint arXiv:2007.03113* (2020).
- [16] Seyed Mehran Kazemi, Anton Tsitsulin, Hossein Esfandiari, MohammadHossein Bateni, Deepak Ramachandran, Bryan Perozzi, and Vahab Mirrokni. 2022. Tackling Provably Hard Representative Selection via Graph Neural Networks. *arXiv:2205.10403* [cs.LG]
- [17] Elan Sopher Markowitz, Keshav Balasubramanian, Mehrnoosh Mirtaheeri, Sami Abu-El-Haija, Bryan Perozzi, Greg Ver Steeg, and Aram Galstyan. 2021. Graph Traversal with Tensor Functionals: A Meta-Algorithm for Scalable Learning. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=6DOZ8XNNfGN>
- [18] John Palowitch and Bryan Perozzi. 2020. Debiasing graph representations via metadata-orthogonal training. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 435–442.
- [19] John Palowitch, Anton Tsitsulin, Brandon Mayer, and Bryan Perozzi. 2022. GraphWorld: Fake Graphs Bring Real Insights for GNNs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (Washington DC, USA) (KDD '22)*. Association for Computing Machinery, New York, NY, USA, 3691–3701. <https://doi.org/10.1145/3534678.3539203>
- [20] Bryan Perozzi. 2016. *Local Modeling of Attributed Graphs: Algorithms and Applications*. Ph. D. Dissertation. State University of New York at Stony Brook.
- [21] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*. 701–710.
- [22] Ștefan Postăvaru, Anton Tsitsulin, Filipe Miguel Gonçalves de Almeida, Yingtao Tian, Silvio Lattanzi, and Bryan Perozzi. 2020. InstantEmbedding: Efficient local node representations. *arXiv preprint arXiv:2010.06992* (2020).
- [23] Benedek Rozemberczki, Peter Englert, Amol Kapoor, Martin Blais, and Bryan Perozzi. 2021. Pathfinder discovery networks for neural message passing. In *Proceedings of the Web Conference 2021*. 2547–2558.
- [24] Anton Tsitsulin, Marina Munkhoeva, and Bryan Perozzi. 2020. Just slaq when you approximate: Accurate spectral distances for web-scale graphs. In *Proceedings of the Web Conference 2020*. 2697–2703.
- [25] Anton Tsitsulin, Marina Munkhoeva, and Bryan Perozzi. 2023. Unsupervised Embedding Quality Evaluation. *arXiv:2305.16562* [cs.LG]
- [26] Anton Tsitsulin, Benedek Rozemberczki, John Palowitch, and Bryan Perozzi. 2022. Synthetic Graph Generation to Benchmark Graph Learning. *arXiv preprint arXiv:2204.01376* (2022).
- [27] Minji Yoon, John Palowitch, Dustin Zelle, Ziniu Hu, Russ Salakhutdinov, and Bryan Perozzi. 2022. Zero-shot Transfer Learning within a Heterogeneous Graph via Knowledge Transfer Networks. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.).
- [28] Qi Zhu, Yizhu Jiao, Haonan Wang, Natalia Ponomareva, and Bryan Perozzi. 2023. GNN Domain Adaptation using Optimal Transport. <https://openreview.net/forum?id=nGyWzq-703u>
- [29] Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. 2021. Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 27965–27977. <https://proceedings.neurips.cc/paper/2021/file/eb55e369affa90f7dd7dc9e2cd33b16-Paper.pdf>