

Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao

Graph Neural Networks

Foundations, Frontiers, and Applications

July 13, 2021

Pre-publication draft of a book to be published by Springer.
Unedited version released with permission. All relevant
copyrights held by the author and publisher extend to this
pre-publication draft.

Preface

The field of graph neural networks (GNNs) has seen rapid and incredible strides over the recent years. Graph neural networks, also known as deep learning on graphs, graph representation learning, or geometric deep learning have become one of the fastest-growing research topics in machine learning, especially deep learning. This wave of research at the intersection of graph theory and deep learning has also influenced other fields of science, including recommendation systems, computer vision, natural language processing, inductive logic programming, program synthesis, software mining, automated planning, cybersecurity, and intelligent transportation.

Although graph neural networks have achieved remarkable attention, it still faces many challenges when applying them into other domains, from the theoretical understanding of methods to the scalability and interpretability in a real system, and from the soundness of the methodology to the empirical performance in an application. However, as the field rapidly grows, it has been extremely challenging to gain a global perspective of the developments of GNNs. Therefore, we feel the urgency to bridge the above gap and have a comprehensive book on this fast-growing yet challenging topic, which can benefit a broad audience including advanced undergraduate and graduate students, postdoctoral researchers, lecturers, and industrial practitioners.

This book is intended to cover a broad range of topics in graph neural networks, from the foundations to the frontiers, and from the methodologies to the applications. Our book is dedicated to introducing the fundamental concepts and algorithms of GNNs, new research frontiers of GNNs, and broad and emerging applications with GNNs.

Book Website and Resources

The website and further resources of this book can be found at: <https://graph-neural-networks.github.io/>. The website provides online preprints and lecture slides of all the chapters. It also provides pointers to useful material and resources that are publicly available and relevant to graph neural networks.

To the Instructors

The book can be used for a one-semester graduate course for graduate students. Though it is mainly written for students with a background in computer science, students with a basic understanding of probability, statistics, graph theory, linear algebra, and machine learning techniques such as deep learning will find it easily accessible. Some chapters can be skipped or assigned as homework assignments for reviewing purposes if students have knowledge of a chapter. For example, if students have taken a deep learning course, they can skip Chapter 1. The instructors can also choose to combine Chapters 1, 2, and 3 together as a background introduction course at the very beginning.

When the course focuses more on the foundation and theories of graph neural networks, the instructor can choose to focus more on Chapters 4-8 while using Chapters 19-27 to showcase the applications, motivations, and limitations. Please refer to the Editors' Notes at the end of each chapter on how Chapters 4-8 and Chapters 19-27 are correlated. When the course focuses more on the research frontiers, Chapters 9-18 can be the pivot to organize the course. For example, the instructor can make it an advanced graduate course where the students are asked to search and present the most recent research papers in each different research frontier. They can also be asked to establish their course project based on the applications described in Chapters 19-27 as well as the materials provided on our website.

To the Readers

This book was designed to cover a wide range of topics in the graph neural network field, including background, theoretical foundations, methodologies, research frontiers, and applications. Therefore, it can be treated as a comprehensive handbook for a wide variety of readers such as students, researchers, and professionals. You should have some knowledge of the concepts and terminology associated with statistics, machine learning, and graph theory. Some backgrounds of the basics have been provided and referenced in the first eight chapters. You should better also have knowledge of deep learning and some programming experience for easily accessing the most of chapters of this book. In particular, you should be able to read pseudocode and understand graph structures.

The book is well modularized and each chapter can be learned in a standalone manner based on the special interests and needs. For those readers who want to have a solid understanding of various techniques and theories of graph neural networks, you can start from Chapters 4-9. For those who further want to perform in-depth research and advance related fields, please read those chapters of interest among Chapters 9-18, which provide comprehensive knowledge in the most recent research issues, open problems, and research frontiers. For those who want to apply graph neural networks to benefit specific domains, or aim at finding interesting applications to validate specific graph neural networks techniques, please refer to Chapters 19-27.

Acknowledgements

Graph machine learning has attracted many gifted researchers to make their seminal contributions over the last few years. We are very fortunate to discuss the challenges and opportunities, and often work with many of them on a rich variety of research topics in this exciting field. We are deeply indebted to these collaborators and colleagues from JD.COM, IBM Research, Tsinghua University, Simon Fraser University, Emory University, and elsewhere, who encouraged us to create such a book comprehensively covering various topics of Graph Neural Networks in order to educate the interested beginners and foster the advancement of the field for both academic researchers and industrial practitioners.

This book would not have been possible without the contributions of many people. We would like to give many thanks to the people who offered feedback on checking the consistency of the math notations of the entire book as well as reference editing of this book. They are people from Emory University: Ling Chen, Xiaojie Guo, and Shiyu Wang, as well as people from Tsinghua University: Yue He, Ziwei Zhang, and Haoxin Liu. We would like to give our special thanks to Dr. Xiaojie Guo, who generously offered her help in providing numerous valuable feedback on many chapters.

We also want to thank those who allowed us to reproduce images, figures, or data from their publications.

Finally, we would like to thank our families for their love, patience and support during this very unusual time when we are writing and editing this book.

Editor Biography



Dr. Lingfei Wu is a Principal Scientist at JD.COM Silicon Valley Research Center, leading a team of 30+ machine learning/natural language processing scientists and software engineers to build intelligent e-commerce personalization system. He earned his Ph.D. degree in computer science from the College of William and Mary in 2016. Previously, he was a research staff member at IBM Thomas J. Watson Research Center and led a 10+ research scientist team for developing novel Graph Neural Networks methods and systems, which leads to the 1 AI Challenge Project in IBM Research and multiple IBM Awards including three-time Outstanding Technical

Achievement Award. He has published more than 90 top-ranked conference and journal papers, and is a co-inventor of more than 40 filed US patents. Because of the high commercial value of his patents, he has received eight invention achievement awards and has been appointed as IBM Master Inventors, class of 2020. He was the recipients of the Best Paper Award and Best Student Paper Award of several conferences such as IEEE ICC'19, AAAI workshop on DLGMA'20 and KDD workshop on DLG'19. His research has been featured in numerous media outlets, including NatureNews, YahooNews, Venturebeat, TechTalks, SyncedReview, Leiphone, QbitAI, MIT News, IBM Research News, and SIAM News. He has co-organized 10+ conferences (KDD, AAAI, IEEE BigData) and is the founding co-chair for Workshops of Deep Learning on Graphs (with AAAI'21, AAAI'20, KDD'21, KDD'20, KDD'19, and IEEE BigData'19). He has currently served as Associate Editor for IEEE Transactions on Neural Networks and Learning Systems, ACM Transactions on Knowledge Discovery from Data and International Journal of Intelligent Systems, and regularly served as a SPC/PC member of the following major AI/ML/NLP conferences including KDD, IJCAI, AAAI, NIPS, ICML, ICLR, and ACL.



Dr. Peng Cui is an Associate Professor with tenure at Department of Computer Science in Tsinghua University. He obtained his PhD degree from Tsinghua University in 2010. His research interests include data mining, machine learning and multimedia analysis, with expertise on network representation learning, causal inference and stable learning, social dynamics modeling, and user behavior modeling, etc. He is keen to promote the convergence and integration of causal inference and machine learning, addressing the fundamental issues of today's AI technology, including explainability, stability and

fairness issues. He is recognized as a Distinguished Scientist of ACM, Distinguished Member of CCF and Senior Member of IEEE. He has published more than 100 papers in prestigious conferences and journals in machine learning and data mining. He is one of the most cited authors in network embedding. A number of his proposed algorithms on network embedding generate substantial impact in academia and industry. His recent research won the IEEE Multimedia Best Department Paper Award, IEEE ICDM 2015 Best Student Paper Award, IEEE ICME 2014 Best Paper Award, ACM MM12 Grand Challenge Multimodal Award, MMM13 Best Paper Award, and were selected into the Best of KDD special issues in 2014 and 2016, respectively. He was PC co-chair of CIKM2019 and MMM2020, SPC or area chair of ICML, KDD, WWW, IJCAI, AAAI, etc., and Associate Editors of IEEE TKDE (2017-), IEEE TBD (2019-), ACM TIST(2018-), and ACM TOMM (2016-) etc. He received ACM China Rising Star Award in 2015, and CCF-IEEE CS Young Scientist Award in 2018.



Dr. Jian Pei is a Professor in the School of Computing Science at Simon Fraser University. He is a well-known leading researcher in the general areas of data science, big data, data mining, and database systems. His expertise is on developing effective and efficient data analysis techniques for novel data intensive applications, and transferring his research results to products and business practice. He is recognized as a Fellow of the Royal Society of Canada (Canada's national academy), the Canadian Academy of Engineering, the Association of Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). He is one of the most cited authors in data mining, database systems, and information retrieval. Since 2000, he has published one textbook, two monographs and over 300 research papers in refereed journals and conferences, which have been cited extensively by others. His research has generated remarkable impact substantially beyond academia. For example, his algorithms have been adopted by industry in production and popular open-source software suites. Jian Pei also demonstrated outstanding professional leadership in many academic organizations and activities. He was the editor-in-chief of the IEEE Transactions of Knowledge and Data Engineering (TKDE) in 2013-16, the chair of the Special Interest Group on Knowledge Discovery in Data (SIGKDD) of the Association for Computing Machinery (ACM) in 2017-2021, and a general co-chair or program committee co-chair of many premier conferences. He maintains a wide spectrum of industry relations with both global and local industry partners. He is an active consultant and coach for industry on enterprise data strategies, healthcare informatics, network security intelligence, computational finance, and smart retail. He received many prestigious awards, including the 2017 ACM SIGKDD Innovation Award, the 2015 ACM SIGKDD Service Award, the 2014 IEEE ICDM Research Contributions Award, the British Columbia Innovation Council 2005 Young Innovator Award, an NSERC 2008 Discovery Accelerator Supplements Award (100 awards cross the whole country), an IBM Faculty Award (2006), a KDD Best Application Paper Award (2008), an ICDE Influential Paper Award (2018), a PAKDD Best Paper Award (2014), a PAKDD Most Influential Paper Award (2009), and an IEEE Outstanding Paper Award (2007).



Dr. Liang Zhao is an assistant professor at the Department of Compute Science at Emory University. Before that, he was an assistant professor in the Department of Information Science and Technology and the Department of Computer Science at George Mason University. He obtained his PhD degree in 2016 from Computer Science Department at Virginia Tech in the United States. His research interests include data mining, artificial intelligence, and machine learning, with special interests in spatiotemporal and network data mining, deep learning on graphs, nonconvex optimization, model parallelism, event prediction, and interpretable machine learning. He received AWS Machine Learning Research Award in 2020 from Amazon Company for his research on distributed graph neural networks. He won NSF Career Award in 2020 awarded by National Science Foundation for his research on deep learning for spatial networks, and Jeffress Trust Award in 2019 for his research on deep generative models for biomolecules, awarded by Jeffress Memorial Trust Foundation and Bank of America. He won the Best Paper Award in the 19th IEEE International Conference on Data Mining (ICDM 2019) for the paper of his lab on deep graph transformation. He has also won Best Paper Award Shortlist in the 27th Web Conference (WWW 2021) for deep generative models. He was selected as “Top 20 Rising Star in Data Mining” by Microsoft Search in 2016 for his research on spatiotemporal data mining. He has also won Outstanding Doctoral Student in the Department of Computer Science at Virginia Tech in 2017. He is awarded as CIFellow Mentor 2021 by the Computing Community Consortium for his research on deep learning for spatial data. He has published numerous research papers in top-tier conferences and journals such as KDD, TKDE, ICDM, ICLR, Proceedings of the IEEE, ACM Computing Surveys, TKDD, IJCAI, AAAI, and WWW. He has been serving as organizers such as publication chair, poster chair, and session chair for many top-tier conferences such as SIGSPATIAL, KDD, ICDM, and CIKM.

List of Contributors

Miltiadis Allamanis
Microsoft Research, Cambridge, UK

Yu Chen
Facebook AI, Menlo Park, CA, USA

Yunfei Chu
Alibaba Group, Hangzhou, China

Peng Cui
Tsinghua University, Beijing, China

Tyler Derr
Vanderbilt University, Nashville, TN, USA

Keyu Duan
Texas A&M University, College Station, TX, USA

Qizhang Feng
Texas A&M University, College Station, TX, USA

Stephan Günnemann
Technical University of Munich, München, Germany

Xiaojie Guo
JD.COM Silicon Valley Research Center, Mountain View, CA, USA

Yu Hou
Weill Cornell Medicine, New York City, New York, USA

Xia Hu
Texas A&M University, College Station, TX, USA

Junzhou Huang
University of Texas at Arlington, Arlington, TA, United States

Shouling Ji

Zhejiang University, Hangzhou, China

Wei Jin

Michigan State University, East Lansing, MI, USA

Anowarul Kabir

George Mason University, Fairfax, VA, USA

Seyed Mehran Kazemi

Borealis AI, Montreal, Canada.

Jure Leskovec

Stanford University, Stanford, CA, USA

Juncheng Li

Zhejiang University, Hangzhou, China

Jiacheng Li

Zhejiang University, Hangzhou, China

Pan Li

Purdue University, Lafayette, IN, USA

Yanhua Li

Worcester Polytechnic Institute, Worcester, MA, USA

Renjie Liao

University of Toronto, Toronto, Canada

Xiang Ling

Zhejiang University, Hangzhou, China

Bang Liu

University of Montreal, Montreal, Canada

Ninghao Liu

Texas A&M University, College Station, TX, USA

Zirui Liu

Texas A&M University, College Station, TX, USA

Hehuan Ma

University of Texas at Arlington, College Station, TX, USA

Collin McMillan

University of Notre Dame, Notre Dame, IN, USA

Christopher Morris

Polytechnique Montréal, Montréal, Canada

Zongshen Mu

Zhejiang University, Hangzhou, China

Menghai Pan

Worcester Polytechnic Institute, Worcester, MA, USA

Jian Pei
Simon Fraser University, British Columbia, Canada

Yu Rong
Tencent AI Lab, Shenzhen, China

Amarda Shehu
George Mason University, Fairfax, VA, USA

Kai Shen
Zhejiang University, Hangzhou, China

Chuan Shi
Beijing University of Posts and Telecommunications, Beijing, China

Le Song
Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates

Chang Su
Weill Cornell Medicine, New York City, New York, USA

Jian Tang
Mila-Quebec AI Institute, HEC Montreal, Canada

Siliang Tang
Zhejiang University, Hangzhou, China

Fei Wang
Weill Cornell Medicine, New York City, New York, USA

Shen Wang
University of Illinois at Chicago, Chicago, IL, USA

Shiyu Wang
Emory University, Atlanta, GA, USA

Xiao Wang
Beijing University of Posts and Telecommunications, Beijing, China

Yu Wang
Vanderbilt University, Nashville, TN, USA

Chunming Wu
Zhejiang University, Hangzhou, China

Lingfei Wu
JD.COM Silicon Valley Research Center, Mountain View, CA, USA

Hongxia Yang
Alibaba Group, Hangzhou, China

Jiangchao Yao

Alibaba Group, Hangzhou, China

Philip S. Yu

University of Illinois at Chicago, Chicago, IL, USA

Muhan Zhang

Peking University, Beijing, China

Wenqiao Zhang

Zhejiang University, Hangzhou, China

Liang Zhao

Emory University, Atlanta, GA, USA

Chang Zhou

Alibaba Group, Hangzhou, China

Kaixiong Zhou

Texas A&M University, TX, USA

Xun Zhou

University of Iowa, Iowa City, IA, USA

Contents

Terminologies	xxix
1 Basic concepts of Graphs	xxix
2 Machine Learning on Graphs	xxx
3 Graph Neural Networks	xxx
Notations	xxxiii
Part I Introduction	
1 Representation Learning	3
Liang Zhao, Lingfei Wu, Peng Cui and Jian Pei	
1.1 Representation Learning: An Introduction	3
1.2 Representation Learning in Different Areas	5
1.2.1 Representation Learning for Image Processing	5
1.2.2 Representation Learning for Speech Recognition	8
1.2.3 Representation Learning for Natural Language Processing	10
1.2.4 Representation Learning for Networks	13
1.3 Summary	14
2 Graph Representation Learning	17
Peng Cui, Lingfei Wu, Jian Pei, Liang Zhao and Xiao Wang	
2.1 Graph Representation Learning: An Introduction	17
2.2 Traditional Graph Embedding	19
2.3 Modern Graph Embedding	20
2.3.1 Structure-Property Preserving Graph Representation Learning	20
2.3.2 Graph Representation Learning with Side Information	23
2.3.3 Advanced Information Preserving Graph Representation Learning	24
2.4 Graph Neural Networks	25
2.5 Summary	26

3	Graph Neural Networks	27
	Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao and Le Song	
3.1	Graph Neural Networks: An Introduction	28
3.2	Graph Neural Networks: Overview	29
3.2.1	Graph Neural Networks: Foundations	29
3.2.2	Graph Neural Networks: Frontiers	31
3.2.3	Graph Neural Networks: Applications	33
3.2.4	Graph Neural Networks: Organization	35
3.3	Summary	36
Part II Foundations of Graph Neural Networks		
4	Graph Neural Networks for Node Classification	41
	Jian Tang and Renjie Liao	
4.1	Background and Problem Definition	41
4.2	Supervised Graph Neural Networks	42
4.2.1	General Framework of Graph Neural Networks	43
4.2.2	Graph Convolutional Networks	44
4.2.3	Graph Attention Networks	46
4.2.4	Neural Message Passing Networks	48
4.2.5	Continuous Graph Neural Networks	48
4.2.6	Multi-Scale Spectral Graph Convolutional Networks	51
4.3	Unsupervised Graph Neural Networks	54
4.3.1	Variational Graph Auto-Encoders	54
4.3.2	Deep Graph Infomax	57
4.4	Over-smoothing Problem	59
4.5	Summary	61
5	The Expressive Power of Graph Neural Networks	63
	Pan Li and Jure Leskovec	
5.1	Introduction	63
5.2	Graph Representation Learning and Problem Formulation	67
5.3	The Power of Message Passing Graph Neural Networks	70
5.3.1	Preliminaries: Neural Networks for Sets	70
5.3.2	Message Passing Graph Neural Networks	71
5.3.3	The Expressive Power of MP-GNN	72
5.3.4	MP-GNN with the Power of the 1-WL Test	75
5.4	Graph Neural Networks Architectures that are more Powerful than 1-WL Test	77
5.4.1	Limitations of MP-GNN	77
5.4.2	Injecting Random Attributes	79
5.4.3	Injecting Deterministic Distance Attributes	86
5.4.4	Higher-order Graph Neural Networks	92
5.5	Summary	97

6 Graph Neural Networks: Scalability	99
Hehuan Ma, Yu Rong, and Junzhou Huang	
6.1 Introduction	100
6.2 Preliminary	101
6.3 Sampling Paradigms	101
6.3.1 Node-wise Sampling	103
6.3.2 Layer-wise Sampling	106
6.3.3 Graph-wise Sampling	111
6.4 Applications of Large-scale Graph Neural Networks on Recommendation Systems	115
6.4.1 Item-item Recommendation	116
6.4.2 User-item Recommendation	116
6.5 Future Directions	118
7 Interpretability in Graph Neural Networks	121
Ninghao Liu and Qizhang Feng and Xia Hu	
7.1 Background: Interpretability in Deep Models	121
7.1.1 Definition of Interpretability and Interpretation	122
7.1.2 The Value of Interpretation	123
7.1.3 Traditional Interpretation Methods	124
7.1.4 Opportunities and Challenges	127
7.2 Explanation Methods for Graph Neural Networks	128
7.2.1 Background	128
7.2.2 Approximation-Based Explanation	130
7.2.3 Relevance-Propagation Based Explanation	134
7.2.4 Perturbation-Based Approaches	135
7.2.5 Generative Explanation	137
7.3 Interpretable Modeling on Graph Neural Networks	138
7.3.1 GNN-Based Attention Models	138
7.3.2 Disentangled Representation Learning on Graphs	141
7.4 Evaluation of Graph Neural Networks Explanations	143
7.4.1 Benchmark Datasets	143
7.4.2 Evaluation Metrics	145
7.5 Future Directions	146
8 Graph Neural Networks: Adversarial Robustness	149
Stephan Günnemann	
8.1 Motivation	149
8.2 Limitations of Graph Neural Networks: Adversarial Examples	152
8.2.1 Categorization of Adversarial Attacks	152
8.2.2 The Effect of Perturbations and Some Insights	156
8.2.3 Discussion and Future Directions	159
8.3 Provable Robustness: Certificates for Graph Neural Networks	160
8.3.1 Model-Specific Certificates	160
8.3.2 Model-Agnostic Certificates	163
8.3.3 Advanced Certification and Discussion	165

8.4	Improving Robustness of Graph Neural Networks	165
8.4.1	Improving the Graph	166
8.4.2	Improving the Training Procedure	167
8.4.3	Improving the Graph Neural Networks' Architecture	170
8.4.4	Discussion and Future Directions	171
8.5	Proper Evaluation in the View of Robustness	172
8.6	Summary	175

Part III Frontiers of Graph Neural Networks

9	Graph Neural Networks: Graph Classification	179
	Christopher Morris	
9.1	Introduction	179
9.2	Graph neural networks for graph classification: Classic works and modern architectures	180
9.2.1	Spatial approaches	181
9.2.2	Spectral approaches	184
9.3	Pooling layers: Learning graph-level outputs from node-level outputs	186
9.3.1	Attention-based pooling layers	187
9.3.2	Cluster-based pooling layers	187
9.3.3	Other pooling layers	188
9.4	Limitations of graph neural networks and higher-order layers for graph classification	189
9.4.1	Overcoming limitations	190
9.5	Applications of graph neural networks for graph classification	191
9.6	Benchmark Datasets	192
9.7	Summary	192
10	Graph Neural Networks: Link Prediction	195
	Muhan Zhang	
10.1	Introduction	195
10.2	Traditional Link Prediction Methods	197
10.2.1	Heuristic Methods	197
10.2.2	Latent-Feature Methods	200
10.2.3	Content-Based Methods	203
10.3	GNN Methods for Link Prediction	203
10.3.1	Node-Based Methods	203
10.3.2	Subgraph-Based Methods	206
10.3.3	Comparing Node-Based Methods and Subgraph-Based Methods	209
10.4	Theory for Link Prediction	211
10.4.1	γ -Decaying Heuristic Theory	211
10.4.2	Labeling Trick	217
10.5	Future Directions	220
10.5.1	Accelerating Subgraph-Based Methods	220

10.5.2	Designing More Powerful Labeling Tricks	221
10.5.3	Understanding When to Use One-Hot Features	222
11	Graph Neural Networks: Graph Generation	225
	Renjie Liao	
11.1	Introduction	225
11.2	Classic Graph Generative Models	226
11.2.1	Erdős–Rényi Model	226
11.2.2	Stochastic Block Model	228
11.3	Deep Graph Generative Models	229
11.3.1	Representing Graphs	230
11.3.2	Variational Auto-Encoder Methods	230
11.3.3	Deep Autoregressive Methods	236
11.3.4	Generative Adversarial Methods	244
11.4	Summary	250
12	Graph Neural Networks: Graph Transformation	251
	Xiaojie Guo, Shiyu Wang, Liang Zhao	
12.1	Problem Formulation of Graph Transformation	252
12.2	Node-level Transformation	253
12.2.1	Definition of Node-level Transformation	253
12.2.2	Interaction Networks	253
12.2.3	Spatio-Temporal Convolution Recurrent Neural Networks	254
12.3	Edge-level Transformation	256
12.3.1	Definition of Edge-level Transformation	256
12.3.2	Graph Transformation Generative Adversarial Networks	257
12.3.3	Multi-scale Graph Transformation Networks	259
12.3.4	Graph Transformation Policy Networks	260
12.4	Node-Edge Co-Transformation	261
12.4.1	Definition of Node-Edge Co-Transformation	261
12.4.2	Editing-based Node-Edge Co-Transformation	266
12.5	Other Graph-based Transformations	271
12.5.1	Sequence-to-Graph Transformation	271
12.5.2	Graph-to-Sequence Transformation	272
12.5.3	Context-to-Graph Transformation	273
12.6	Summary	275
13	Graph Neural Networks: Graph Matching	277
	Xiang Ling, Lingfei Wu, Chunming Wu and Shouling Ji	
13.1	Introduction	278
13.2	Graph Matching Learning	279
13.2.1	Problem Definition	280
13.2.2	Deep Learning based Models	282
13.2.3	Graph Neural Network based Models	284
13.3	Graph Similarity Learning	288
13.3.1	Problem Definition	288

13.3.2	Graph-Graph Regression Tasks	290
13.3.3	Graph-Graph Classification Tasks	293
13.4	Summary	295
14	Graph Neural Networks: Graph Structure Learning	297
	Yu Chen and Lingfei Wu	
14.1	Introduction	297
14.2	Traditional Graph Structure Learning	299
14.2.1	Unsupervised Graph Structure Learning	299
14.2.2	Supervised Graph Structure Learning	301
14.3	Graph Structure Learning for Graph Neural Networks	303
14.3.1	Joint Graph Structure and Representation Learning	304
14.3.2	Connections to Other Problems	317
14.4	Future Directions	319
14.4.1	Robust Graph Structure Learning	319
14.4.2	Scalable Graph Structure Learning	320
14.4.3	Graph Structure Learning for Heterogeneous Graphs	320
14.5	Summary	320
15	Dynamic Graph Neural Networks	323
	Seyed Mehran Kazemi	
15.1	Introduction	323
15.2	Background and Notation	325
15.2.1	Graph Neural Networks	325
15.2.2	Sequence Models	327
15.2.3	Encoder-Decoder Framework and Model Training	330
15.3	Categories of Dynamic Graphs	331
15.3.1	Discrete vs. Continues	331
15.3.2	Types of Evolution	333
15.3.3	Prediction Problems, Interpolation, and Extrapolation	334
15.4	Modeling Dynamic Graphs with Graph Neural Networks	335
15.4.1	Conversion to Static Graphs	335
15.4.2	Graph Neural Networks for DTDGs	337
15.4.3	Graph Neural Networks for CTDGs	340
15.5	Applications	343
15.5.1	Skeleton-based Human Activity Recognition	343
15.5.2	Traffic Forecasting	345
15.5.3	Temporal Knowledge Graph Completion	346
15.6	Summary	348
16	Heterogeneous Graph Neural Networks	351
	Chuan Shi	
16.1	Introduction to HGNNs	351
16.1.1	Basic Concepts of Heterogeneous Graphs	353
16.1.2	Challenges of HG Embedding	354
16.1.3	Brief Overview of Current Development	355

16.2	Shallow Models	356
16.2.1	Decomposition-based Methods	357
16.2.2	Random Walk-based Methods	358
16.3	Deep Models	360
16.3.1	Message Passing-based Methods (HGNNs)	360
16.3.2	Encoder-decoder-based Methods	363
16.3.3	Adversarial-based Methods	364
16.4	Review	366
16.5	Future Directions	367
16.5.1	Structures and Properties Preservation	367
16.5.2	Deeper Exploration	367
16.5.3	Reliability	368
16.5.4	Applications	369
17	Graph Neural Networks: AutoML	371
	Kaixiong Zhou, Zirui Liu, Keyu Duan and Xia Hu	
17.1	Background	372
17.1.1	Notations of AutoGNN	373
17.1.2	Problem Definition of AutoGNN	375
17.1.3	Challenges in AutoGNN	375
17.2	Search Space	376
17.2.1	Architecture Search Space	377
17.2.2	Training Hyperparameter Search Space	380
17.2.3	Efficient Search Space	381
17.3	Search Algorithms	382
17.3.1	Random Search	382
17.3.2	Evolutionary Search	382
17.3.3	Reinforcement Learning Based Search	383
17.3.4	Differentiable Search	385
17.3.5	Efficient Performance Estimation	386
17.4	Future Directions	387
18	Graph Neural Networks: Self-supervised Learning	391
	Yu Wang, Wei Jin, and Tyler Derr	
18.1	Introduction	392
18.2	Self-supervised Learning	393
18.3	Applying SSL to Graph Neural Networks: Categorizing Training Strategies, Loss Functions and Pretext Tasks	395
18.3.1	Training Strategies	396
18.3.2	Loss Functions	399
18.3.3	Pretext Tasks	402
18.4	Node-level SSL Pretext Tasks	403
18.4.1	Structure-based Pretext Tasks	403
18.4.2	Feature-based Pretext Tasks	404
18.4.3	Hybrid Pretext Tasks	406
18.5	Graph-level SSL Pretext Tasks	408

18.5.1	Structure-based Pretext Tasks	408
18.5.2	Feature-based Pretext Tasks	413
18.5.3	Hybrid Pretext Tasks	414
18.6	Node-graph-level SSL Pretext Tasks	417
18.7	Discussion	418
18.8	Summary	419

Part IV Broad and Emerging Applications with Graph Neural Networks

19	Graph Neural Networks in Modern Recommender Systems	423
	Yunfei Chu, Jiangchao Yao, Chang Zhou and Hongxia Yang	
19.1	Graph Neural Networks for Recommender System in Practice	423
19.1.1	Introduction	423
19.1.2	Classic Approaches to Predict User-Item Preference	428
19.1.3	Item Recommendation in user-item Recommender Systems: a Bipartite Graph Perspective	429
19.2	Case Study 1: Dynamic Graph Neural Networks Learning	431
19.2.1	Dynamic Sequential Graph	431
19.2.2	DSGL: Dynamic Sequential Graph Learning	432
19.2.3	Model Prediction	435
19.2.4	Experiments and Discussions	436
19.3	Case Study 2: Device-Cloud Collaborative Learning for Graph Neural Networks	438
19.3.1	The proposed framework	438
19.3.2	Experiments and Discussions	442
19.4	Future Directions	444
20	Graph Neural Networks in Computer Vision	447
	Siliang Tang, Wenqiao Zhang, Zongshen Mu, Kai Shen, Juncheng Li, Jiacheng Li and Lingfei Wu	
20.1	Introduction	448
20.2	Representing Vision as Graphs	448
20.2.1	Visual Node representation	448
20.2.2	Visual Edge representation	450
20.3	Case Study 1: Image	451
20.3.1	Object Detection	451
20.3.2	Image Classification	453
20.4	Case Study 2: Video	454
20.4.1	Video Action Recognition	454
20.4.2	Temporal Action Localization	456
20.5	Other Related Work: Cross-media	457
20.5.1	Visual Caption	457
20.5.2	Visual Question Answering	458
20.5.3	Cross-Media Retrieval	459
20.6	Frontiers for Graph Neural Networks on Computer Vision	460
20.6.1	Advanced Graph Neural Networks for Computer Vision	460

20.6.2	Broader Area of Graph Neural Networks on Computer Vision	461
20.7	Summary	462
21	Graph Neural Networks in Natural Language Processing	463
	Bang Liu, Lingfei Wu	
21.1	Introduction	463
21.2	Modeling Text as Graphs	466
21.2.1	Graph Representations in Natural Language Processing	466
21.2.2	Tackling Natural Language Processing Tasks from a Graph Perspective	468
21.3	Case Study 1: Graph-based Text Clustering and Matching	470
21.3.1	Graph-based Clustering for Hot Events Discovery and Organization	470
21.3.2	Long Document Matching with Graph Decomposition and Convolution	473
21.4	Case Study 2: Graph-based Multi-Hop Reading Comprehension	475
21.5	Future Directions	479
21.6	Conclusions	480
22	Graph Neural Networks in Program Analysis	483
	Miltiadis Allamanis	
22.1	Introduction	483
22.2	Machine Learning in Program Analysis	484
22.3	A Graph Representation of Programs	486
22.4	Graph Neural Networks for Program Graphs	489
22.5	Case Study 1: Detecting Variable Misuse Bugs	491
22.6	Case Study 2: Predicting Types in Dynamically Typed Languages	493
22.7	Future Directions	495
23	Graph Neural Networks in Software Mining	499
	Collin McMillan	
23.1	Introduction	499
23.2	Modeling Software as a Graph	500
23.2.1	Macro versus Micro Representations	501
23.2.2	Combining the Macro- and Micro-level	503
23.3	Relevant Software Mining Tasks	503
23.4	Example Software Mining Task: Source Code Summarization	504
23.4.1	Primer GNN-based Code Summarization	505
23.4.2	Directions for Improvement	510
23.5	Summary	512
24	GNN-based Biomedical Knowledge Graph Mining in Drug Development	517
	Chang Su, Yu Hou, Fei Wang	
24.1	Introduction	517

24.2	Existing Biomedical Knowledge Graphs	518
24.3	Inference on Knowledge Graphs	523
24.3.1	Conventional KG inference techniques	523
24.3.2	GNN-based KG inference techniques	524
24.4	KG-based hypothesis generation in computational drug development	528
24.4.1	A machine learning framework for KG-based drug repurposing	529
24.4.2	Application of KG-based drug repurposing in COVID-19	530
24.5	Future directions	531
24.5.1	KG quality control	532
24.5.2	Scalable inference	533
24.5.3	Coupling KGs with other biomedical data	533
25	Graph Neural Networks in Predicting Protein Function and Interactions	541
	Anowarul Kabir and Amarda Shehu	
25.1	From Protein Interactions to Function: An Introduction	541
25.1.1	Enter Stage Left: Protein-Protein Interaction Networks	542
25.1.2	Problem Formulation(s), Assumptions, and Noise: A Historical Perspective	543
25.1.3	Shallow Machine Learning Models over the Years	543
25.1.4	Enter Stage Right: Graph Neural Networks	544
25.2	Highlighted Case Studies	547
25.2.1	Case Study 1: Prediction of Protein-Protein and Protein-Drug Interactions: The Link Prediction Problem	547
25.2.2	Case Study 2: Prediction of Protein Function and Functionally-important Residues	549
25.2.3	Case Study 3: From Representation Learning to Multirelational Link Prediction in Biological Networks with Graph Autoencoders	553
25.3	Future Directions	555
26	Graph Neural Networks in Anomaly Detection	557
	Shen Wang, Philip S. Yu	
26.1	Introduction	557
26.2	Issues	561
26.2.1	Data-specific issues	561
26.2.2	Task-specific Issues	563
26.2.3	Model-specific Issues	563
26.3	Pipeline	564
26.3.1	Graph Construction and Transformation	564
26.3.2	Graph Representation Learning	565
26.3.3	Prediction	567
26.4	Taxonomy	568
26.5	Case Studies	568

26.5.1	Case Study 1: Graph Embeddings for Malicious Accounts Detection	569
26.5.2	Case Study 2: Hierarchical Attention Mechanism based Cash-out User Detection	570
26.5.3	Case Study 3: Attentional Heterogeneous Graph Neural Networks for Malicious Program Detection	572
26.5.4	Case Study 4: Graph Matching Framework to Learn the Program Representation and Similarity Metric via Graph Neural Networks for Unknown Malicious Program Detection	573
26.5.5	Case Study 5: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN	575
26.5.6	Case Study 6: GCN-based Anti-Spam for Spam Review Detection	576
26.6	Future Directions	577
27	Graph Neural Networks in Urban Intelligence	579
	Yanhua Li, Xun Zhou, and Menghai Pan	
27.1	Graph Neural Networks for Urban Intelligence	580
27.1.1	Introduction	580
27.1.2	Application scenarios in urban intelligence	581
27.1.3	Representing urban systems as graphs	584
27.1.4	Case Study 1: Graph Neural Networks in urban configuration and transportation	586
27.1.5	Case Study 2: Graph Neural Networks in urban anomaly and event detection	588
27.1.6	Case Study 3: Graph Neural Networks in urban human behavior inference	589
27.1.7	Future Directions	591
	References	593

Terminologies

This chapter describes a list of definitions of terminologies related to graph neural networks used throughout this book.

1 Basic concepts of Graphs

- **Graph:** A graph is composed of a node set and an edge set, where nodes represent entities and edges represent the relationship between entities. The nodes and edges form the topology structure of the graph. Besides the graph structure, nodes, edges, and/or the whole graph can be associated with rich information represented as node/edge/graph features (also known as attributes or contents).
- **Subgraph:** A subgraph is a graph whose set of nodes and set of edges are all subsets of the original graph.
- **Centrality:** A centrality is a measurement of the importance of nodes in the graph. The basic assumption of centrality is that a node is thought to be important if many other important nodes also connect to it. Common centrality measurements include the degree centrality, the eigenvector centrality, the betweenness centrality, and the closeness centrality.
- **Neighborhood:** The neighborhood of a node generally refers to other nodes that are close to it. For example, the k -order neighborhood of a node, also called the k -step neighborhood, denotes a set of other nodes in which the shortest path distance between these nodes and the central node is no larger than k .
- **Community Structure:** A community refers to a group of nodes that are densely connected internally and less densely connected externally.
- **Graph Sampling:** Graph sampling is a technique to pick a subset of nodes and/or edges from the original graph. Graph sampling can be applied to train machine learning models on large-scale graphs while preventing severe scalability issues.

- **Heterogeneous Graphs:** Graphs are called heterogeneous if the nodes and/or edges of the graph are from different types. A typical example of heteronomous graphs is knowledge graphs where the edges are composed of different types.
- **Hypergraphs:** Hypergraphs are generalizations of graphs in which an edge can join any number of nodes.
- **Random Graph:** Random graph generally aims to model the probability distributions over graphs that the observed graphs are generated from. The most basic and well-studied random graph model, known as the Erdos–Renyi model, assumes that the node set is fixed and each edge is identically and independently generated.
- **Dynamic Graph:** Dynamic graph refers to when at least one component of the graph data changes over time, e.g., adding or deleting nodes, adding or deleting edges, changing edges weights or changing node attributes, etc. If graphs are not dynamic, we refer to them as static graphs.

2 Machine Learning on Graphs

- **Spectral Graph Theory:** Spectral graph theory analyzes matrices associated with the graph such as its adjacency matrix or Laplacian matrix using tools of linear algebra such as studying the eigenvalues and eigenvectors of the matrix.
- **Graph Signal Processing:** Graph Signal Processing (GSP) aims to develop tools for processing signals defined on graphs. A graph signal refers to a finite collection of data samples with one sample at each node in the graph.
- **Node-level Tasks:** Node-level tasks refer to machine learning tasks associated with individual nodes in the graph. Typical examples of node-level tasks include node classification and node regression.
- **Edge-level Tasks:** Edge-level tasks refer to machine learning tasks associated with a pair of nodes in the graph. A typical example of an edge-level task is link prediction.
- **Graph-level Tasks:** Graph-level tasks refer to machine learning tasks associated with the whole graph. Typical examples of graph-level tasks include graph classification and graph property prediction.
- **Transductive and Inductive Learning:** Transductive learning refers to that the targeted instances such as nodes or edges are observed at the training time (though the labels of the targeted instances remain unknown) and inductive learning aims to learn the model which is generalizable to unobserved instances.

3 Graph Neural Networks

- **Network embedding:** The goal of network embedding is to represent each node in the graph as a low-dimensional vector so that useful information such as the

graph structures and some properties of the graph is preserved in the embedding vectors. Network embedding is also referred to as graph embedding and node representation learning.

- **Graph Neural Network:** Graph neural network refers to any neural network working on the graph data.
- **Graph Convolutional Network:** Graph convolutional network usually refers to a specific graph neural network proposed by Kipf and Welling [Kipf and Welling \(2017a\)](#). It is occasionally used as a synonym for graph neural network, i.e., referring to any neural network working on the graph data, in some literature.
- **Message-Passing:** Message-passing is a framework of graph neural networks in which the key step is to pass messages between different nodes based on graph structures in each neural network layer. The most widely adopted formulation, usually denoted as message-passing neural networks, is to only pass messages between nodes that are directly connected [Gilmer et al \(2017\)](#). The message passing functions are also called graph filters and graph convolutions in some literature.
- **Readout:** Readout refers to functions that summarize the information of individual nodes to form more high-level information such as forming a subgraph/super-graph or obtaining the representations of the entire graph. Readout is also called pooling and graph coarsening in some literature.
- **Graph Adversarial Attack:** Graph adversarial attacks aim to generate worst-case perturbations by manipulating the graph structure and/or node features so that the performance of some models are downgraded. Graph adversarial attacks can be categorized based on the attacker's goals, capabilities, and accessible knowledge.
- **Robustness certificates:** Methods providing formal guarantees that the prediction of a GNN is not affected even when perturbations are performed based on a certain perturbation model.

Notations

This Chapter provides a concise reference that describes the notations used throughout this book.

Numbers, Arrays, and Matrices

A scalar	x
A vector	\mathbf{x}
A matrix	X
An identity matrix	\mathbf{I}
The set of real numbers	\mathbb{R}
The set of complex numbers	\mathbb{C}
The set of integers	\mathbb{Z}
The set of real n -length vectors	\mathbb{R}^n
The set of real $m \times n$ matrices	$\mathbb{R}^{m \times n}$
The real interval including a and b	$[a, b]$
The real interval including a but excluding b	$[a, b)$
The element of the vector \mathbf{x} with index i	\mathbf{x}_i
The element of matrix X 's indexed by Row i and Column j	$X_{i,j}$

Graph Basics

A graph	\mathcal{G}
Edge set	\mathcal{E}
Vertex set	\mathcal{V}
Adjacent matrix of a graph	A
Laplacian matrix	L
Diagonal degree matrix	D
Isomorphism between graphs \mathcal{G} and \mathcal{H}	$\mathcal{G} \cong \mathcal{H}$
\mathcal{H} is a subgraph of graph \mathcal{G}	$\mathcal{H} \subseteq \mathcal{G}$
\mathcal{H} is a proper subgraph of graph \mathcal{G}	$\mathcal{H} \subset \mathcal{G}$
Union of graphs \mathcal{H} and \mathcal{G}	$\mathcal{G} \cup \mathcal{H}$

Intersection of graphs \mathcal{H} and \mathcal{G}	$\mathcal{G} \cap \mathcal{H}$
Disjoint Union of graphs \mathcal{H} and \mathcal{G}	$\mathcal{G} + \mathcal{H}$
Cartesian Product of graphs of graphs \mathcal{H} and \mathcal{G}	$\mathcal{G} \times \mathcal{H}$
The join of graphs \mathcal{H} and \mathcal{G}	$\mathcal{G} \vee \mathcal{H}$

Basic Operations

Transpose of matrix X	X^\top
Dot product of matrices X and Y	$X \cdot Y$ or XY
Element-wise (Hadamard) product of matrices X and Y	$X \odot Y$
Determinant of X	$\det(X)$
p -norm (also called ℓ_p norm) of \mathbf{x}	$\ \mathbf{x}\ _p$
Union	\cup
Intersection	\cap
Subset	\subseteq
Proper subset	\subset
Inner product of vector \mathbf{x} and \mathbf{y}	$\langle \mathbf{x}, \mathbf{y} \rangle$

Functions

The function f with domain \mathbb{A} and range \mathbb{B}	$f : \mathbb{A} \rightarrow \mathbb{B}$
Derivative of y with respect to \mathbf{x}	$\frac{dy}{d\mathbf{x}}$
Partial derivative of y with respect to \mathbf{x}	$\frac{\partial y}{\partial \mathbf{x}}$
Gradient of y with respect to \mathbf{x}	$\nabla_{\mathbf{x}} y$
Matrix derivatives of y with respect to matrix X	$\nabla_X y$
The Hessian matrix of function f at input vector \mathbf{x}	$\nabla^2 f(\mathbf{x})$
Definite integral over the entire domain of \mathbf{x}	$\int f(\mathbf{x}) d\mathbf{x}$
Definite integral with respect to \mathbf{x} over the set \mathbb{S}	$\int_{\mathbb{S}} f(\mathbf{x}) d\mathbf{x}$
A function of \mathbf{x} parametrized by θ	$f(\mathbf{x}; \theta)$
Convolution between functions f and g	$f * g$

Probabilistic Theory

A probability distribution of a	$p(a)$
A conditional probabilistic distribution of b given a	$p(b a)$
The random variables a and b are independent	$a \perp b$
Variables a and b are conditionally independent given c	$a \perp b \mid c$
Random variable a has a distribution p	$a \sim p$
The expectation of $f(a)$ with respect to the variable a under distribution p	$\mathbb{E}_{a \sim p}[f(a)]$
Gaussian distribution over \mathbf{x} with mean $\boldsymbol{\mu}$ and covariance Σ	$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$